

T.C.
MARMARA ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ
EKONOMETRİ ANABİLİM DALI
İSTATİSTİK BİLİM DALI

**DERİN ÖĞRENME ALGORİTMALARI İLE HİSSE
SENETLERİNİN FİYAT HAREKETLİLİĞİ
ÖNGÖRÜSÜ**

Doktora Tezi

Ceren CAMKIRAN

İstanbul, 2023

T.C.
MARMARA ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ
EKONOMETRİ ANABİLİM DALI
İSTATİSTİK BİLİM DALI

**DERİN ÖĞRENME ALGORİTMALARI İLE HİSSE
SENETLERİNİN FİYAT HAREKETLİLİĞİ
ÖNGÖRÜSÜ**

Doktora Tezi

Ceren CAMKIRAN

Danışman: Prof. Dr. Dilek ALTAŞ KARACA

İstanbul, 2023

GENEL BİLGİLER

Adı ve Soyadı	: Ceren Camkıran
Anabilim Dalı	: Ekonometri
Programı	: İstatistik
Tez Danışmanı	: Prof. Dr. Dilek Altaş Karaca
Tez Türü ve Tarihi	: Doktora - Aralık, 2023
Anahtar Kelimeler	: Derin Öğrenme, CNN, LSTM, GRU, Fiyat Hareketliliği

ÖZET

DERİN ÖĞRENME ALGORİTMALARI İLE HİSSE SENETLERİNİN FİYAT HAREKETLİLİĞİ ÖNGÖRÜSÜ

Bilgi teknolojilerindeki gelişmelerle ortaya çıkan çeşitli türdeki büyük verilerin analizi için kullanılan yöntemler arasında, sinir ağlarını temel alan derin öğrenme algoritmaları dikkat çekmektedir. Finans alanında kullanılan derin öğrenme yöntemleri ile yatırım modelleri, ileriye yönelik fiyat tahmini, formasyon sınıflandırma ve al – sat karar modelleri gibi yatırımcıya fikir verebilecek modeller geliştirilmektedir. Bu çalışmada, derin öğrenme algoritmaları kullanılarak BİST30’da yer alan hisse senetleri için mum grafiklerinden hareketle gelecek dönem fiyat ve fiyat hareketliliği tahmini yapılması amaçlanmıştır. Bu amaçla, fiyat eğiliminde dönüşüm bildiren 8 adet üçlü mum formasyonu belirlenmiştir. Mum formasyonlarının görüntülerine ait hazır veri seti olmadığından, etiketleme işlemi otomatik yapılarak etiketli bir veri seti oluşturulmuştur. Etiketlenen mum formasyonu görüntüleri için Konvolüsyon Sinir Ağı (CNN) ile bir görüntü sınıflandırma modeli geliştirilmiştir. Buradan elde edilen her bir formasyona atanma olasılıkları, daha sonra fiyat tahmini için kullanılan Uzun – Kısa Süreli Hafıza Ağları (LSTM) ve Geçitli Yinelemeli Birimler (GRU) algoritmalarına girdi olarak alınmıştır. Yalnızca bu olasılıkların yer aldığı veri setine ek olarak açılış, kapanış, en düşük/en yüksek değer ve işlem hacmi bilgilerinin yer aldığı ikinci bir veri seti oluşturulmuş, ardından bu iki veri seti birleştirilerek hibrit bir veri seti elde edilmiştir. İki algoritmanın performansı tüm veri setleri için karşılaştırmalı olarak değerlendirilmiştir. Yapılan fiyat tahmininde tüm veri setleri için LSTM modeli daha başarılı bulunmuştur. Hibrit veri setinden elde edilen fiyat tahmin modeli diğer veri setlerine göre daha düşük hata oranına sahiptir. Fiyatların gelecek dönem hareket yönleri incelendiğinde, olasılık bilgisini içeren iki modelin hareket yönlerini daha yüksek oranda doğru tahmin ettikleri belirlenmiştir. Böylece görüntü tabanlı formasyon bilgisinin fiyat tahmininde ve hareket yönü belirlemede etkili olduğu tespit edilmiştir.

GENERAL KNOWLEDGE

Name and Surname : Ceren Camkıran
Field : Econometrics
Programme : Statistics
Supervisor : Prof. Dr. Dilek Altaş Karaca
Degree Awarded and Date : PhD - December, 2023
Keywords : Deep Learning, CNN, LSTM, GRU, Price Movement

ABSTRACT

PREDICTION OF STOCK PRICE MOVEMENTS WITH DEEP LEARNING ALGORITHMS

Among the methods used to analyze various types of big data that emerge with developments in information technologies, deep learning algorithms based on neural networks attract attention. Deep learning methods employed in the financial sector aim to develop models that can provide insights to investors, including investment models, price forecastings, pattern classification, and buy-sell decision models. This study focuses on predicting future prices and price movements for stocks listed in BIST30 using deep learning algorithms based on candlestick charts. To achieve this goal, eight triple candlestick patterns indicating reversals in price trends were identified. Since there was no pre-labeled dataset for the images of candlestick patterns, the labeling process was automated to create a labeled dataset. A convolutional neural network (CNN) algorithm was then used to develop an image classification model for the labeled candlestick pattern images. The probabilities assigned to each identified pattern were later used as input for LSTM (Long Short-Term Memory) and GRU (Gated Recurrent Unit) algorithms employed for price prediction. In addition to the dataset containing only these probabilities, a second dataset was created, including information such as opening, closing, low/high values, and trading volume. These two datasets were merged to obtain a hybrid dataset. The performance of the two algorithms was evaluated comparatively across all datasets. The LSTM model was found to be more successful in price prediction for all datasets. The price prediction model derived from the hybrid dataset exhibited a lower error rate compared to other datasets. When examining the future movement directions of prices, it was determined that the models incorporating probability information more accurately predicted price movement directions. Thus, it was concluded that image-based pattern information is effective in price prediction and determining price movement directions.

İÇİNDEKİLER

Sayfa No.

TABLO LİSTESİ	vi
ŞEKİL LİSTESİ	vii
KISALTMALAR.....	x
1. GİRİŞ	1
2. DERİN ÖĞRENME	4
2.1 Derin Öğrenmenin Tarihsel Gelişimi	5
2.2 Derin Öğrenme Yaklaşımları.....	7
2.2.1 Denetimli Öğrenme.....	7
2.2.2 Denetimsiz Öğrenme.....	8
2.2.3 Yarı-Denetimli Öğrenme.....	8
2.2.4 Takviyeli Öğrenme.....	8
2.3 Derin Öğrenme Yöntemlerinin Özellikleri.....	9
2.4 Uygulama Alanları	11
2.5 Derin Öğrenmede Kullanılan Kavramlar	12
2.5.1 Nöron	12
2.5.2 Ağırlıklar	13
2.5.3 Sapma (Bias) ve Toplanma Fonksiyonu	13
2.5.4 Girdi Katmanı - Çıktı Katmanı - Gizli Katman	13
2.5.5 Aktivasyon Fonksiyonları	14
2.5.6 İleri Besleme ve Geri Yayılım.....	23
2.5.7 Kayıp - Maliyet Fonksiyonu	26
2.5.7.1 Sınıflandırma İçin Kullanılan Kayıp Fonksiyonları.....	27
2.5.7.2 Regresyon İçin Kullanılan Kayıp Fonksiyonları.....	32
2.6 Optimizasyon Algoritmaları	36
2.6.1 Gradyan İniş Algoritması	36
2.6.2 Yığın Gradyan İniş.....	38
2.6.3 Stokastik Gradyan İniş	39
2.6.4 Mini - Yığın Gradyan İniş.....	39

2.6.5	Uyarlanabilir Moment Tahmini (ADAM).....	40
2.7	Eksik - Aşırı Öğrenme, Varyans - Sapma Dengesi, Regülerizasyon.....	42
2.7.1	Çapraz Geçerlilik (Doğrulama).....	45
2.7.2	Regülerizasyon Teknikleri	46
3.	DERİN ÖĞRENME ALGORİTMALARI	49
3.1	Konvolüsyonel Sinir Ağları.....	49
3.1.1	CNN Mimarisi.....	50
3.1.2	Konvolüsyon Katmanı	54
3.1.3	Havuzlama Katmanı.....	58
3.1.4	Düzleştirme ve Tam Bağlantılı Katman.....	60
3.1.5	Toplu Normalleştirme ve Seyreltme	61
3.1.6	Sınıflandırma Katmanı.....	62
3.2	Diğer Konvolüsyon Sinir Ağı Mimarileri	63
3.2.1	LeNet-5	63
3.2.2	AlexNet	64
3.2.3	VGGNet	65
3.2.4	GoogleNet	66
3.2.5	ResNet	68
3.3	Yinelemeli Sinir Ağları	69
3.3.1	RNN Mimarisi.....	70
3.3.1.1	Jordan Ağları	73
3.3.1.2	Elman Ağları	74
3.3.2	Yinelemeli Sinir Ağlarında Geri Yayılım.....	75
3.3.3	Gradyan Kaybı ve Patlaması.....	77
3.3.4	Uzun - Kısa Süreli Hafıza Ağları (LSTM).....	78
3.3.5	Geçitli Yinelemeli Birimler (GRU)	81
4.	BİST30 MUM FORMASYONU SINIFLANDIRMASI VE FİYAT HAREKETLİLİĞİ ÖNGÖRÜSÜ	84
4.1	Teknik Analiz	84
4.1.1	Destek ve Direnç	87
4.1.2	Grafikler	88
4.1.2.1	Çizgi Grafiği.....	89
4.1.2.2	Çubuk Grafiği.....	89

4.1.2.3	Nokta ve Şekil (X-O) Grafiği.....	90
4.1.2.4	Kagi Grafiği.....	91
4.1.2.5	Mum Grafiği.....	92
4.1.3	Formasyonlar	93
4.1.3.1	Trend Formasyonları	94
4.1.3.2	Mum Formasyonları	94
4.2	Literatür Taraması	100
4.3	Veri Seti ve Metodoloji	105
4.4	Bulgular.....	110
4.4.1	CNN Bulguları	110
4.4.2	LSTM ve GRU Bulguları.....	121
5.	SONUÇ	134
EKLER		138
KAYNAKÇA		158

TABLO LİSTESİ

Tablo 1:	Kayıp Fonksiyonları.....	27
Tablo 2:	Hisse Senedi Kodları.....	105
Tablo 3:	CNN Modeli Çıktı Yapısı	113
Tablo 4:	AKBNK Verilerinin Sonuçları	116
Tablo 5:	Hisse Senetlerinin Test Verisi Sonuçları	117
Tablo 6:	Formasyonlar için Karmaşıklık (Confusion) Matrisi.....	120
Tablo 7:	Kullanılan LSTM - GRU Mimarileri	125
Tablo 8:	Olasılık Veri Seti için LSTM ve GRU Model Yapıları	126
Tablo 9:	Fiyat Veri Seti için LSTM ve GRU Model Yapıları.....	126
Tablo 10:	Olasılık_Fiyat Veri Seti için LSTM ve GRU Model Yapıları.....	127
Tablo 11:	YKBNK Model Sonuçları.....	128
Tablo 12:	YKBNK Tahmin ve Hareket Yönü Sonuçları	129
Tablo 13:	BRMEN Model Sonuçları.....	130
Tablo 14:	BRMEN Tahmin ve Hareket Yönü Sonuçları	130
Tablo 15:	SAHOL Model Sonuçları	131
Tablo 16:	SAHOL Tahmin ve Hareket Yönü Sonuçları	132

ŞEKİL LİSTESİ

Şekil 1: Yapay Zeka, Makine Öğrenmesi ve Derin Öğrenme İlişkisi.....	4
Şekil 2: Biyolojik Nöron Yapısı.....	12
Şekil 3: Yapay Nöronun İşleyişi	12
Şekil 4: Çok Katmanlı Bir İleri Beslemeli Yapay Sinir Ağı	14
Şekil 5: Aktivasyon Fonksiyonlarının Performans Karşılaştırılması	15
Şekil 6: Basamak Fonksiyonu ve Türevi.....	16
Şekil 7: Doğrusal Fonksiyon ve Türevi.....	16
Şekil 8: Sigmoid Fonksiyonu ve Türevi.....	17
Şekil 9: Tanh Fonksiyonu ve Türevi	18
Şekil 10: ReLU Fonksiyonu ve Türevi.....	19
Şekil 11: Sızıntı ReLU Fonksiyonu ve Türevi	20
Şekil 12: Softplus Fonksiyonu ve Türevi	21
Şekil 13: ELU Fonksiyonu ve Türevi ($\alpha=1$ için).....	22
Şekil 14: Swish ReLU Fonksiyonu ve Türevi.....	22
Şekil 15: Bir Gizli Katmanlı Geri Yayılım Algoritması	23
Şekil 16: Hinge Kayıp Fonksiyonu	31
Şekil 17: Sınıflandırma Kayıp Fonksiyonlarının Grafikleri.....	32
Şekil 18: Karesel Kayıp ve Mutlak Kayıp Karşılaştırılması	33
Şekil 19: Regresyon Kayıp Fonksiyonları.....	35
Şekil 20: Kantil Kayıp Fonksiyonu	35
Şekil 21: Gradyan İniş Algoritması.....	37
Şekil 22: Gradyan İniş Algoritması Grafiği	37
Şekil 23: Gradyan İniş Algoritmaları için Yörüngeler	38
Şekil 24: Artan Model Karmaşıklığı Aşırı Öğrenme	42
Şekil 25: Eksik Öğrenme ve Aşırı Öğrenme	43
Şekil 26: K-katlı Çapraz Geçerlilik	46
Şekil 27: CNN Mimarisi Genel Yapısı.....	51
Şekil 28: Konvolüsyon İşlemi	56
Şekil 29: Dolgu İşlemi.....	57

Şekil 30: ReLU Aktivasyon Fonksiyonu.....	58
Şekil 31: Maksimum, Ortalama ve Minimum Havuzlama Örneği	59
Şekil 32: Düzleştirme İşlemi ve Tam Bağlantılı Katman.....	61
Şekil 33: Standart Sinir Ağı (a) ve Seyreltme İşlemi Yapılan Sinir Ağı (b)	62
Şekil 34: LeNet-5 Mimarisi.....	64
Şekil 35: AlexNet Mimarisi	65
Şekil 36: VGGNet Mimarisi.....	66
Şekil 37: Başlangıç (inception) Modülü.....	67
Şekil 38: GoogleNet Mimarisi	67
Şekil 39: Artık Blok Yapısı	68
Şekil 40: ResNet Mimarisi	68
Şekil 41: Yinelemeli Sinir Ağı Döngüsü ve Açılmış Gösterimi	70
Şekil 42: Çeşitli RNN Mimarileri	72
Şekil 43: Jordan Ağları.....	73
Şekil 44: Elman Ağları	74
Şekil 45: Sigmoid Fonksiyonu ve Gradyanı.....	78
Şekil 46: LSTM Hücresi.....	79
Şekil 47: GRU Hücresi.....	81
Şekil 48: Destek-Direnç Seviyeleri	88
Şekil 49: Çizgi Grafiği	89
Şekil 50: Çubuk Grafiği	90
Şekil 51: Nokta ve Şekil (X-O) Grafiği.....	91
Şekil 52: Kagi Grafiği	91
Şekil 53: Mum Grafiğinin Temel Yapısı.....	92
Şekil 54: Mum Grafiği	93
Şekil 55: Çekiç ve Asılı Adam Mum Formasyonları	95
Şekil 56: Yukarı Yutan ve Aşağı Yutan Mum Formasyonları	95
Şekil 57: Yukarı Harami ve Aşağı Harami Mum Formasyonları	96
Şekil 58: Sabah Yıldızı ve Akşam Yıldızı Mum Formasyonları.....	97
Şekil 59: Üç Yıldız Aşağı ve Üç Yıldız Yukarı Mum Formasyonları	97
Şekil 60: Üç İçeride Yukarı ve Üç İçeride Aşağı Mum Formasyonları	98
Şekil 61: Üç Dıştan Yukarı ve Üç Dıştan Aşağı Mum Formasyonları	99

Şekil 62: Yükselen Üçler ve Düşen Üçler Mum Formasyonları.....	99
Şekil 63: Etiketli Mum Formasyonları Klasörleri.....	107
Şekil 64: Eğitim Veri Setindeki Formasyonların Dağılımı	111
Şekil 65: Geçerlilik Veri Setindeki Formasyonların Dağılımı	112
Şekil 66: Eğitim ve Geçerlilik İçin Doğruluk Oranı	114
Şekil 67: Eğitim ve Geçerlilik için Kayıp Fonksiyonu	115
Şekil 68: YKBNK Hisse Senedi Mum Grafiği	122
Şekil 69: BRMEN Hisse Senedi Mum Grafiği	123
Şekil 70: SAHOL Hisse Senedi Mum Grafiği	124



KISALTMALAR

ADAM	Uyarlanabilir Moment Tahmini
AI	Yapay Zeka
BİST	Borsa İstanbul
CAE	Evrişimsel Otomatik Kodlayıcı
CNN	Konvolüsyonel Sinir Ağları (Convolutional Neural Networks)
GRU	Geçitli Yinelemeli Birimler
HPC	High Performance Computing (Yüksek Performanslı Hesaplama)
LBANN	Livermore Big Artificial Neural Network (Livermore Büyük Yapay Sinir Ağları)
LSTM	Uzun - Kısa Süreli Hafıza Ağları
NLP	Doğal Dil İşleme
NN	Sinir Ağları
RNN	Yinelemeli Sinir Ağları
YSA	Yapay Sinir Ağları

1. GİRİŞ

Teknolojide yaşanan gelişmeler, dijitalleşmenin giderek artması, internet ve bilgi sistemleri kullanımının yaygınlaşması ile birlikte üretilen veri miktarı önemli ölçüde artmış, verinin yapısında farklılaşmalar yaşanmış ve verinin farklı türleri analiz edilebilir hale gelmiştir. Video, ses, resim, metin vb. veri türleri alışlagelmiş veri türlerine alternatif olarak ortaya çıkmıştır. Veride yaşanan bu değişimi yakalamakta geleneksel yöntemler yetersiz kalmış ve verinin yapısına uygun yöntemler geliştirilmiştir. Bu yöntemler arasında veriden öğrenmeye dayalı makine öğrenmesi ve derin öğrenme algoritmaları öne çıkmaktadır. Bu algoritmalar, yüksek teknolojik altyapıya sahip olan bilgisayarların gelişmesiyle, büyük miktarda ve yüksek boyuttaki verilere erişimi ve bu verilerin işlenmesini mümkün kılmıştır. Bu algoritmalarla sınıflandırma, öngörü, tahmin, nesne tanıma, doğal dil işleme gibi problemlere çözüm üretilmektedir. Makine öğrenmesi ve derin öğrenme algoritmaları, yapay zeka teknolojileri, endüstri, tıp, savunma sanayi, siber güvenlik, meteoroloji gibi pek çok alanda uygulanmaktadır. Bu uygulama alanlarından biri de finans alanıdır.

Yatırımcılar için yatırım ortamlarını ve piyasayı tanımak kadar finansal araçların takibi de önem arz etmektedir. Yatırım kararlarının alınmasında finansal tahmin ve öngörülere ihtiyaç duyulmaktadır. Bu tahmin ve öngörüler, bireylerin ve kurumların doğru yatırımlarla kâr etmesine; beklenmedik piyasa şoklarından da kaçınmalarına imkan sağlamaktadır. Bu amaçlarla, hisse senedi gibi finansal araçlar için ileriye yönelik öngörü, al – sat karar modelleri, portföy yönetimi gibi uygulamalar yapılmaktadır. Bu uygulamalarda zaman serisi analizleri, temel analiz ve teknik analiz yöntemleri kullanılmaktadır.

Teknik analiz finansal piyasa takibinde ve yatırımcıların finansal davranışlarını belirlemede etkin bir biçimde kullanılan bir yöntem olup, geçmiş fiyatların makroekonomik ve sektörel bilgileri içerdiği varsayımıyla hareket etmektedir. Ayrıca teknik analizdeki bir diğer varsayım, geçmiş dönemlerin fiyat hareketliliklerinin gelecek dönemde tekrar edeceğidir. Buradan hareketle çeşitli indikatörler ve grafiklerden yararlanılarak yatırımcıların geleceğe yönelik bilgi edinmesi amaçlanmaktadır. Teknik

analizde sıklıkla kullanılan grafik türlerinden bir tanesi mum grafikleridir. Mum grafikleri ilgili zaman frekansına göre fiyatların en düşük, en yüksek, açılış ve kapanış değerlerini tek bir yapıda gösterme özelliğine sahiptir. Böylece bu grafikler kullanılarak fiyata dair farklı bilgiler bir arada işlenebilmektedir. Bu bilgiler zaman içerisinde birbirine benzer yapılar oluşturmakta ve fiyat hareketleri bakımından ortak karakteristikler sunmaktadır. Bu karakteristik yapılar formasyon olarak bilinen şekillerle temsil edilmekte ve genel piyasa gidişatı hakkında fikir vermektedir. Formasyonların bazıları piyasa trendindeki devamlılığa işaret ederken, bazıları ise değişim olacağına dair sinyaller vermektedir. Trend değişiminden kastedilen durum artış eğilimindeki fiyatın düşüş eğilimine dönmesi veya bunun tam tersi bir dönüşün olmasıdır. Bir başka ifadeyle, finansal terminolojide boğa piyasasından ayı piyasasına veya bunun tersi yönde geçişlerin yaşanmasıdır.

Mum formasyonları fiyat hareketlerine dair önemli bilgiler içermekle birlikte, bu formasyonların sayısal veriden manuel olarak belirlenmesi her zaman kolay değildir. Ayrıca fiyat grafiklerinden elde edilen görüntülerin hangi formasyon yapısına ait olduğunu belirlemek de uzman olmayan yatırımcılar açısından zor olabilmektedir. Bu noktada görüntü verilerinin işlenebildiği derin öğrenme algoritmaları ile mum grafikleri gibi görseller rahatlıkla analizlere dahil edilebilmektedir. Bu sebeple, derin öğrenme algoritmaları hisse senedi fiyat tahmini ve fiyat hareketliliklerinin incelenmesi gibi finansal analizlerde geleneksel olarak kullanılan yöntemlere güçlü bir alternatif oluşturmuştur.

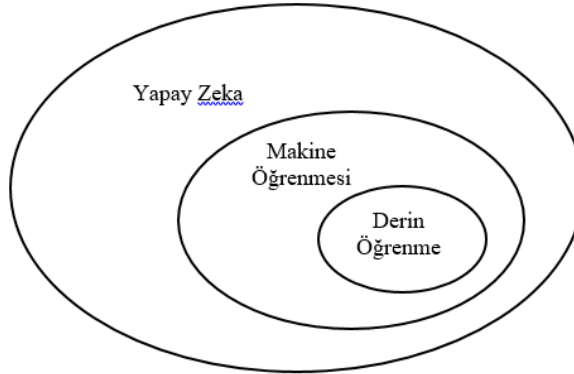
Bu çalışmada, derin öğrenme algoritmaları kullanılarak, Borsa İstanbul'da işlem gören BİST30 hisse senetleri için fiyat hareketliliği tahmini yapılması amaçlanmıştır. Bu amaçla, dönüşüm bildiren bazı önemli mum formasyonlarından görüntü sınıflandırma yapılarak formasyonlar tespit edilmiş ve ardından tespit edilen formasyonlarla ileriye yönelik fiyat tahmini ve fiyat hareketliliği tahminleri elde edilmiştir. Bu doğrultuda öncelikle ilgilenilen formasyonlar için etiketli görüntüler oluşturulmuştur. Ardından formasyonların sınıflandırılması için derin öğrenme yöntemlerinden Konvolüsyon Sinir Ağları (CNN) kullanılmıştır. Buradan elde edilen sınıflandırma bilgisi fiyatların gelecek değerlerini ve gelecek dönem hareket yönlerini tahmin etmekte kullanılmıştır. Ancak burada her bir formasyona ait olma bilgisi yerine, özgün bir yaklaşımla formasyonların

her birine atanma olasılıkları analize dahil edilmiştir. Bu olasılıklar ile fiyatların açılış, kapanış, en düşük, en yüksek ve işlem hacmi değerleri bir arada ve ayrı ayrı kullanılarak ileriye yönelik tahmin yapılmıştır. Tahmin işlemi için derin öğrenme algoritmalarından Uzun – Kısa Süreli Hafıza Ağları (LSTM) ve Geçitli Yinelemeli Birimler (GRU) kullanılmıştır. Böylece formasyonların olasılık bilgisinin gelecek dönem fiyat hareketliliğinin tahmininde ne kadar etkili olduğu karşılaştırmalı olarak incelenmiştir.

Çalışmanın ikinci bölümünde derin öğrenmenin gelişiminden ve özelliklerinden söz edilmiştir. Bölümün devamında derin öğrenme terminolojisinde kullanılan kavramlara ve bazı optimizasyon algoritmalarına dair açıklamalara yer verilmiştir. Üçüncü bölümde temel derin öğrenme algoritmalarından CNN ve türevleri ile birlikte Yinelemeli Sinir Ağları (RNN), LSTM ve GRU'nun detayları sunulmuştur. Dördüncü bölümde ise BİST30 hisse senetleri için yapılan uygulama yer almaktadır. Bu bölümde öncelikle uygulamaya konu olan teknik analiz kavramlarına ve ilgili literatüre yer verilmiştir. Ardından veri etiketleme, sınıflandırma ve tahmin için kullanılan algoritmalarından elde edilen bulgular sunulmuştur. Sonuç bölümünde analiz sonuçları değerlendirilmiş, sonraki bilimsel çalışmalara ve geliştirilen model ile ilgilenebilecek yatırımcılara yönelik önerilere yer verilmiştir.

2. DERİN ÖĞRENME

20. yüzyılın ortalarından itibaren insan zekasına dayalı işlerin makinelerle yaptırılabilmesi fikri ortaya çıkmıştır. Bu fikrin gelişmesi ile birlikte yapay zeka kavramı dikkat çeken bir kavram haline gelmiştir. Yapay zeka (AI), insan zekasına ihtiyaç duyan görevlerin makineler tarafından yapay olarak gerçekleştirmesini sağlayan sistemleri içeren bir bilgisayar bilimi dalıdır. Yapay zeka sistemlerinde ön kurallar ve algoritmalar yer almaktadır. Tanımlanan görevin bu kural ve algoritmalara bağlı kalınarak tamamlanması amaçlanmaktadır. Yapay zekadaki ilerlemeler ile birlikte makine öğrenmesi (ML) ve derin öğrenme (DL) kavramları ortaya çıkmaya başladı. Bu ifadeler zaman zaman birbirleri yerine kullanılmalarına rağmen aynı anlama gelmemektedir. Yapay zeka, makine öğrenmesi ve derin öğrenmeyi de içeren daha geniş kapsamlı bir ifadedir. Her üç kavramın ilişkisi Şekil 1’de yer almaktadır.



Şekil 1: Yapay Zeka, Makine Öğrenmesi ve Derin Öğrenme İlişkisi

Kaynak: François Chollet, “Deep Learning With Python”, New York: Manning Publications Co, 2018, s.4

Yapay zekanın makine öğrenmesi ve derin öğrenme kapsamına girmeyen kısmı veriden öğrenmeye dayalı olmayan sistemlerdir. Başka bir ifade ile salt yapay zeka, kuralların açık bir şekilde tanımlandığı sistemlerdir. Veriden öğrenme aşamasını içeren yapay zeka yaklaşımlarına ise makine öğrenmesi denmektedir. Makineler veriden aldığı bilgileri kullanarak karar vermeyi öğrenmektedir. Daha çok veri daha iyi bir öğrenmeyi beraberinde getireceğinden, makine öğrenmesi dinamik yapıli sistemlerdir. Makine öğrenmesindeki öğrenme işleminde, algoritmadaki hataların azaltılması dolayısıyla

tahmin doğruluğunun artması hedeflenmektedir. Makine öğrenmesi, güçlü bir yapay zeka sisteminin çalışması için kullanılan teknikler bütünüdür.

Derin öğrenme ise, makine öğrenmesinin bir alt kümesi olup beyinde yer alan sinir ağlarının çalışma prensibini temel almaktadır. Beynin yeni bir bilgiyi öğrenmesi sürecine benzer olarak, katmanlardan oluşan aşamalı bir öğrenme yapısı ile veriden özellik çıkarımı yapılmaktadır. İlerleyen katmanlarda bu özellikler birleştirilerek daha karmaşık sistemlerin çözümlenmesi sağlanmaktadır. Burada yer alan derin terimi katman sayısını ifade etmekte ve katman sayısı arttıkça ağın derinliği artmaktadır (Jakhar ve Kaur, 2020).

Makine öğrenmesi ve derin öğrenme arasındaki temel fark, özelliklerin belirlenme sürecidir. Makine öğrenmesi yaklaşımlarında manuel olarak tasarlanan özellikler kullanılarak öğrenme algoritmalarına geçilir. Derin öğrenmede ise özellikler otomatik olarak öğrenilmekte ve katmanlara bağlı olarak hiyerarşik bir şekilde temsil edilmektedir. Bu kısım derin öğrenmenin makine öğrenmesi yaklaşımlarına karşı avantajlı tarafıdır. Diğer taraftan derin öğrenme teknikleri öğrenmenin sağlanabilmesi için büyük miktarda veri gerektirir. Ayrıca tekniklerin uygulanabilmesi için ciddi bir donanım alt yapısına ihtiyaç duymaktadır (Alom ve diğerleri, 2019).

2.1 Derin Öğrenmenin Tarihsel Gelişimi

1950'lerden günümüze, yapay zekanın alt kümesi olarak gelişmeye devam eden makine öğrenmesi, son yıllarda oldukça önemli yeniliklere imkan sağlamaktadır. Sinir ağları (NN), makine öğrenmesinin alt alanı olup, derin öğrenmeyi ortaya çıkaran temellerdendir. Sinir ağlarının temelleri genel olarak nörobiyolojik çalışmalara dayanmaktadır. Bu alanda yapılan çalışmalarda, canlıların sinir sisteminin işleyişi incelenmiş ve bu yapı makineler için yapay öğrenme modelinin temelini oluşturmuştur. İnsan beyninin davranışını taklit etmeye çalışan yapay nöronlar yapay sinir ağı oluşturmak için temel bileşendir. Sinir sisteminde yer alan nöronlar ve bu nöronlar arasındaki bağlantıları temsil eden yapay birimler oluşturulmuştur. Bu sistemin gelişimi çeşitli aşamalardan geçerek birikimli olarak ortaya çıkmıştır. Sinir ağlarının tarihsel

gelişimi aşağıdaki gibi özetlenebilir (Mehrotra ve diğerleri, 1996, s. 4; Alom ve diğerleri, 2019):

- 1938: Rashevsky, sinir ağlarının matematiksel temellerini diferansiyel denklemlerle ifade etmeye katkı sunan nörodinamik çalışmaları başlattı. Bu yaklaşım, daha sonra 1948 yılında Wiener tarafından genişletilmiştir.
- 1943: McCulloch ve Pitts tarafından biyolojik nöronlar için ilk yapay model ortaya konduktan sonra, Landahl, McCulloch ve Pitts (1943), bu modelleri içeren yöntemlerle çeşitli aritmetiksel ve mantıksal işlemlerin gerçekleştirilebileceğini göstermişlerdir.
- 1949: Hebb öğrenme kuralını sunduğu kitabını yayınlamıştır. Burada tekrarlı aktivasyonların bilginin akışına etkisini ortaya koymuştur.
- 1956: Taylor tarafından, Hebb'in kuralına dayalı ilişkisel bellek ağı tanıtılmıştır.
- 1958: Rosenblatt tarafından McCulloch ve Pitts'in nöron modeli için bir öğrenme yöntemi olan algılayıcı (perceptron) kavramı literatüre kazandırılmıştır.
- 1960: Widrow ve Hoff, hata kareler ortalamasını minimize etmek için gradyan iniş kuralı kullanılarak eğitilen “Adaline” ağını tanıtmıştır.
- 1962: Hubel ve Wiesel tarafından kedilerin görsel korteksindeki nöronların özelliklerine ilişkin yapılan biyolojik çalışmalar, kendi kendini eğiten yapay sinir ağı modellerinin gelişmesi için temel oluşturmuşlardır.
- 1969: Minsky ve Papert, uzun bir süre boyunca sinir ağlarındaki araştırmalar üzerine algılayıcıların sınırlarını göstermiştir.

Minsky ve Papert (1969) tarafından bu sınırların ortaya konmasının ardından, yapay zeka çalışmalarına olan ilgi bir süre azalmıştır. Ancak geri yayılım algoritmasının gelişmesi ile beraber çalışmalar yeniden hız kazanmıştır.

- 1985: Hinton ve arkadaşları tarafından geri yayılım algoritması bulunmuştur.
- 1988: Fukushima tarafından, neocognitron görsel örüntü tanıma yeteneğine sahip hiyerarşik bir sinir ağı elde edilmiştir.

- 1989: LeCun ve arkadaşları tarafından el yazısı ile yazılan rakamları okumak için evrişimli sinir ağıları (CNN) kullanıldı, daha sonra 1998’de LeCun ve arkadaşları tarafından belge analizi için CNN’ler geri yayılım algoritması ile birlikte kullanılmıştır.

CNN’in yaygın olarak kullanılmaya başlanması ile birlikte görüntü işleme algoritmalarına verilen önem artmıştır. Bunun yanı sıra farklı veri türlerini (görüntü, metin, video, ses vb.) analiz etmek için çeşitli derin öğrenme algoritmaları geliştirilmektedir.

2.2 Derin Öğrenme Yaklaşımları

Derin öğrenme kavramında yer alan “öğrenme” ifadesi, sinir ağlarının veriyi kullanarak öğrenmesini ve bilgi çıkarımı yapmasını ifade etmektedir. Bu öğrenme aşaması ilgilenilen problemin çözümünün genelleştirilebilmesi için önemli bir aşamadır. Öğrenme aşamasında verinin türüne ve problemin amacına göre çeşitli yaklaşımlar mevcuttur. Derin öğrenme algoritmaları genellikle bu yaklaşımlardan biri baz alınarak uygulanmaktadır. Derin öğrenme yaklaşımları denetimli (supervised), denetimsiz (unsupervised), yarı denetimli (semi-supervised) ve destekleyici (reinforcement) öğrenme olarak sıralanabilir.

2.2.1 Denetimli Öğrenme

Denetimli öğrenme, ilgilenilen problem için her girdiye karşılık bir çıktının mevcut olduğu bu girdi-çıkı verilerinin etiketli bir biçimde makineye aktarıldığı öğrenme yaklaşımıdır. Burada amaç girdi ve çıktı arasındaki ilişki yapısını belirleyen ve minimum hata oranına sahip eşleştirme fonksiyonunu öğrenilmesini sağlamaktır. Algoritmanın parametreleri, düşük bir kayıp oranı ile yüksek bir öğrenme performansı elde edilinceye kadar tekrarlı olarak değiştirilir ve böylece ağı öğrenmesi gerçekleşir. Öğrenme işlemi tamamlandığında, algoritma yeni girdi verilerine karşılık çıktıların hesaplanmasında kullanılabilir (Armoogum ve Li, 2019).

2.2.2 Denetimsiz Öğrenme

Denetimsiz öğrenme, veri etiketleri olmadan öğrenme işleminin gerçekleştirildiği yaklaşımdır. Burada ağ, dışarıdan bir destek almadan girdiler arasındaki benzerlikler ve farklılıklara göre gruplama yapma amacı taşır. Bu durumda aracı olarak kullanılan ağ yapısı, girdi verilerindeki bilinmeyen ilişkileri veya yapıyı keşfetmek için önemli özellikleri öğrenir. Denetimsiz öğrenme yaklaşımlarında, etiketlenmiş çıktı verileri bulunmadığından, gerçek çıktı verileri ile karşılaştırma yapmak söz konusu değildir. Genellikle kümeleme ve boyut indirgeme teknikleri denetimsiz öğrenme yaklaşımları olarak kabul edilir (Da Silva ve diğerleri, 2017, s.26).

2.2.3 Yarı-Denetimli Öğrenme

Denetimli ve denetimsiz öğrenme arasındaki önemli farklardan biri etiketli verinin var olup olmaması durumudur. Etiketli verinin var olduğu, ancak yeterli miktarda olmadığı durumda yarı denetimli öğrenme yaklaşımı kullanılabilir. Burada, veri etiketlemenin yüksek maliyeti vb. durumlardan kaynaklı olarak az miktarda etiketli veri bulunmaktadır. Buna karşılık tahmin edilmek istenen veri sayısı etiketli veriye göre daha fazladır. Bu nedenle öğrenme aşamasında etiketli ve etiketsiz veriler birlikte kullanılır. Bu özelliği ile denetimli ve denetimsiz öğrenme yaklaşımlarının karışımı olarak da bilinmektedir (Kızılkaya ve Oğuzlar, 2018).

2.2.4 Takviyeli Öğrenme

Takviyeli öğrenmede, eğitim için kullanılacak bir çıktı veri kümesi bulunmamaktadır. Buna karşılık eylemden öğrenme durumu söz konusudur. Takviyeli öğrenmenin üç temel bileşeni; öğrenme ajanı, ortam ve eylemlerdir. Öğrenme ajanı ortamla etkileşim kurarak eylemlerde bulunur ve bu eylemlerin doğru veya yanlış olduğunu kaydeden bir sinyal üretilir. Böylece doğru olan eyleme karşılık ödüllendirme sistemi ile ağ eğitilir. Ödülün maksimize edilmesi amaçlanarak, yanlış karar sinyalleri doğru sinyallere dönüşene kadar ağı eğitimi devam eder (Armoogum ve Li, 2019). Optimize edilmesi gereken çok sayıda parametre varsa derin takviyeli öğrenme yaklaşımları etkili bir çözüm sunmaktadır. Buna karşılık, daha az parametrenin olduğu

durumda türev içermeyen bir takviyeli öğrenme yaklaşımı kullanmak daha uygun olacaktır (Alom ve diğerleri, 2019).

2.3 Derin Öğrenme Yöntemlerinin Özellikleri

Derin öğrenme yöntemlerinin çeşitli uygulama alanlarında ve farklı veri türlerinde uygulanabilir olması belirli özellikleri beraberinde getirir. Bu özellikler şu şekildedir:

- Evrensel Öğrenme Yaklaşımı: Hemen hemen her uygulama alanına uygulanabilir olması sebebiyle derin öğrenme yaklaşımı evrensel öğrenme olarak adlandırılmaktadır.
- Robust: Tam olarak tasarlanmış özelliklere ihtiyaç duymamakta ve mevcut veriye ait optimum özelliklerin otomatik olarak öğrenilmesine olanak tanımaktadır. Bu sebeple, girdi verilerinin doğal varyasyonlarına karşı öğrenmenin gerçekleşmesini sağlayan sağlam bir yaklaşımdır.
- Genelleme: Aynı derin öğrenme yaklaşımı ile eğitilmiş bir model, farklı uygulamalarda veya farklı veri türlerinde kullanılabilir. Bu yaklaşım genellikle transfer öğrenimi olarak da bilinmektedir. İncelenen problemin yeterli mevcut veriye sahip olmadığı durumlarda yararlanılan bir yaklaşımdır.
- Ölçeklenebilirlik: Derin öğrenme algoritma yapılarına çok sayıda katman ve nöron eklenebildiğinden ölçeklenebilirlik özelliğine sahiptir. Örneğin, Microsoft tarafından sunulan ResNet derin öğrenme algoritması 1202 katmana sahip bir ağıdır (Van Essen ve diğerleri, 2015).

Derin öğrenme yöntemleri kendine özgü mimarilerini veya hiyerarşik öğrenme yaklaşımlarını kullanan makine öğrenmesinin sinir ağları ile özelleştirilmiş bölümünü oluşturmaktadır. Öğrenme, öğrenilen modelin (algoritmanın) belirli bir görevi yerine getirebilmesi için model parametrelerinin tahmin edilmesinden oluşan kurallı işlemler sürecinin tamamıdır. Örneğin, Yapay Sinir Ağlarında (YSA) parametreler ağırlık matrisleridir. Öte yandan derin öğrenme, özellik öğrenme ve örüntü sınıflandırmada kullanılan, hiyerarşik mimarilere sahip doğrusal olmayan bilgi işlem birimlerinin birçok aşamasının mevcut olmasına izin veren, girdi ve çıktı katmanları arasında yer alan çeşitli

katmanlardan oluşur (Schmidhuber, 2015; Bengio ve diğerleri, 2015). Verilerin gösterimlerine dayalı öğrenme yöntemleri, gösterim (representation) öğrenme olarak da tanımlanabilir (Bengio ve diğerleri, 2013). Son çalışmalar baz alındığında derin öğrenme tabanlı gösterim öğrenme yaklaşımı ile yüksek seviyeli kavramlar düşük seviyeli kavramlardan tanımlanabilmekte, ayrıca tam tersi tanımlamaların da yapılabildiği özellik veya kavramlar hiyerarşisini içermektedir. Bazı çalışmalarda derin öğrenme, farklı uygulama alanları da dahil olmak üzere çok çeşitli sorunlara çözüm üretebilen evrensel bir öğrenme yaklaşımı olarak tanımlanmaktadır. Başka bir ifade ile derin öğrenme, özel bir sorun çözümüne özgü olmayıp gerek kullanım alanı gerekse kullanım amacı bakımından oldukça geniş bir hakimiyet imkanı tanımaktadır (Bengio, 2009).

Derin öğrenmenin dezavantajlı olabileceği durumlar olarak,

- Derin öğrenme kullanarak büyük veri analitiği,
- Derin öğrenme yaklaşımlarının ölçeklenebilirliği,
- Sistemi öğrenmek için verilerin mevcut olmadığı durumlarda önemli olan verileri üretme yeteneği (özellikle ters grafikler gibi bilgisayarla görme görevi için),
- Mobil zeka, FPGA'lar vb. dahil özel amaçlı cihazlar için enerji verimli teknikler,
- Çoklu görev ve aktarım öğrenme veya çoklu modül öğrenme (farklı alanlardan veya farklı modellerle birlikte öğrenme),
- Öğrenmede nedensellikte başa çıkmak

gibi durumlar öne çıkmaktadır. Ancak bu zorluklar için yapılmış ve yapılmakta olan çalışmalar mevcuttur. İlk olarak, büyük veri analitiği zorluğu için, 2014 yılında yapılmış bir çalışma vardır (Chen ve Lin, 2014). Çalışmada, büyük veri probleminin hacmi, hızı, çeşitliliği ve doğruluğu dahil olmak üzere derin öğrenmenin farklı kriterlerle nasıl başa çıkabileceğinin ayrıntıları açıklanmıştır. Zhou ve arkadaşları (2014) ve Najafabadi ve arkadaşları (2015) çalışmalarında büyük veri problemleriyle uğraşırken derin öğrenme yaklaşımlarının farklı avantajlarını da göstermişlerdir. Geleneksel makine öğrenmesi yaklaşımlarının performansının daha az miktarda girdi verisi için daha iyi performans göstermektedir. Ancak veri miktarı arttıkça, geleneksel makine öğrenimi yaklaşımlarının

performansı sabit hale gelirken, derin öğrenme yaklaşımlarının performansı veri miktarındaki artışa göre artış göstermektedir.

Ölçekleme kaynaklı sorunlar için Yüksek Performanslı Hesaplama (HPC) sistemi (süper hesaplama, küme, bulut bilişim) uygulanmaktadır. Veriler hız, çeşitlilik, doğruluk ve hacim olarak artarken, kurumsal sunucuları ve depolamayı kullanarak ölçeklendirme yapmak zorlaşmaktadır. Lawrence Livermore Ulusal Laboratuvarı (LLNL), derin öğrenme için büyük ölçekli uygulama (süper hesaplama ölçeğinde) için Livermore Büyük Yapay Sinir Ağları (LBANN) adı verilen ve açıkça derin öğrenmenin ölçeklenebilirliği sorunu için çerçeve geliştirmiştir.

Veriden nedensel bir modelin nasıl çıkarılabileceğini tanımlayan grafiksel bir model olan nedenselliğe sahip bir öğrenme sistemi sunulmuştur. Son zamanlarda bu tür problemleri çözmek için derin öğrenme tabanlı bir yaklaşım önerilmiştir (Singh ve diğerleri, 2017). Bu çalışmada, sadece verilerin dağılım grafiğine bakarak ikili nedenselliği tahmin etmek için CNN kullanma fikri önerilmiş ve daha az eğitim verisi olduğunda daha iyi performansa sahip olduğu tespit edilmiştir.

2.4 Uygulama Alanları

Derin öğrenme yöntemleri, makinenin eğitilmesinin yararlı olacağı çeşitli uygulama alanlarında kullanılmaktadır. Bu alanlar,

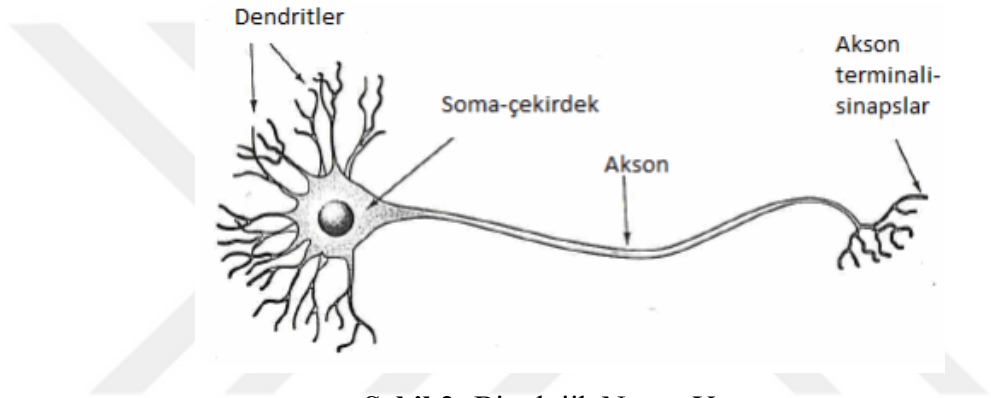
- Uzman bir insan olmaması ve otonom sistemler (Örneğin, Mars'ta navigasyon),
- Konuşma tanıma ve dil anlama,
- Sorunun çözümünün zamanla değişim göstermesi (takip, hava durumu tahmini, tercih, stok, fiyat tahmini),
- Çözümler belirli durumlara (biyometri, kişiselleştirme) uyarlanmak istendiğinde,
- Problemin boyutunun, sınırlı düşünce yeteneğimiz için çok büyük olması (web sayfası sıralamalarını hesaplama, reklamları Facebook ile eşleştirme, duyarlılık analizi)

durumlarında derin öğrenme yaklaşımlarına ihtiyaç duyulmaktadır. Şu anda derin öğrenme hemen hemen her alanda uygulanmaktadır. Sonuç olarak, bu yaklaşıma genellikle evrensel bir öğrenme yaklaşımı denir (Alom ve diğerleri, 2019).

2.5 Derin Öğrenmede Kullanılan Kavramlar

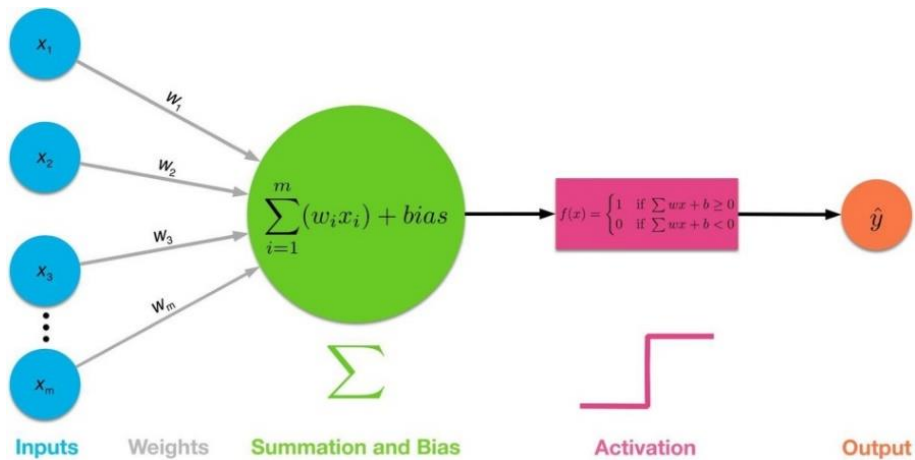
2.5.1 Nöron

Sinir ağları, biyolojik sinir hücrelerinden yola çıkılarak oluşturulmuş yapay sinir hücrelerinden meydana gelmektedir.



Şekil 2: Biyolojik Nöron Yapısı

Nöron birden çok girdinin işlenerek tek bir çıktıya dönüştürüldüğü bir beyin hücresidir. Beyinde bir hücrenin çıktısı, diğer bir hücrenin girdisi olacak şekilde birbirlerine bağlı nöronlardan oluşmaktadır. Nöronlar, dikkate değer hesaplama ve karar alma özelliklerine sahip bir sinir ağı oluşturmaktadırlar.



Şekil 3: Yapay Nöronun İşleyişi

Yapay nöron temel olarak girdiler, ağırlıklar, toplama fonksiyonu, aktivasyon fonksiyonu ve çıktılar olmak üzere 5 kısımdan oluşmaktadır.

2.5.2 Ağırlıklar

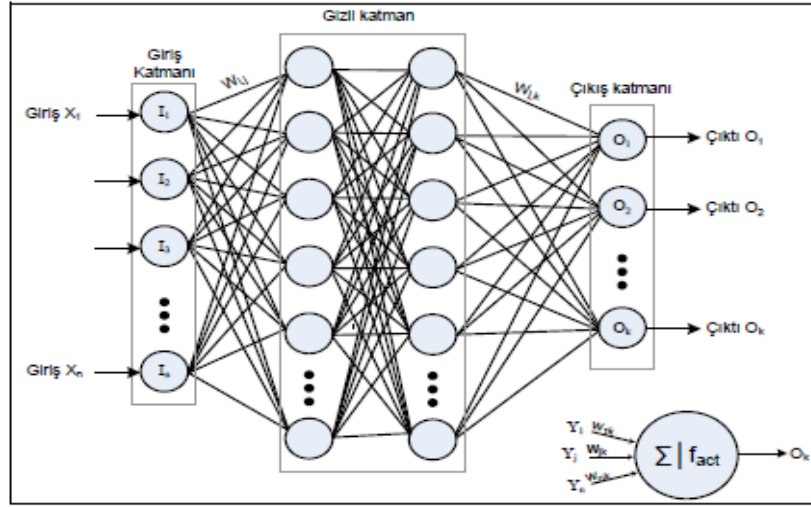
Girdi, nöronda bir ağırlık ile çarpılır. Birden fazla girişi olan bir nöronda her bir nöron girişi o girişe atanan bir ağırlığa sahiptir. Rassal olarak başlatılan bu ağırlıklar, model eğitim süreci içerisinde yenilenmektedir. Eğitim sonrası önemi yüksek olduğu düşünülen girdilere daha yüksek ağırlık değeri verilmektedir. Sıfır ağırlık değeri verilen bir girdinin önemsiz olduğu düşünülebilir (Yılmaz ve Kaya, 2019, s.7).

2.5.3 Sapma (Bias) ve Toplanma Fonksiyonu

Sapma, girdiye ağırlıkla birlikte ek olarak uygulanan doğrusal bileşendir. Girdinin ağırlık ile çarpımı sonucuna ağırlık katsayılı girdinin aralığını değiştirmek için eklenmektedir. Sapma eklenmesi gerçekleştikten sonra elde edilen dönüşüm son doğrusal bileşen olmaktadır. Örneğin X_1 girdisi için w_1 ağırlığı kullanılarak b sapması eklenmiş doğrusal bileşen $X_1w_1 + b$ olarak ifade edilmektedir. Toplanma fonksiyonu, hücredeki net girdiyi hesaplamaktadır. Her bir girdi için elde edilen doğrusal bileşenlerin birleştirilmesi işlemidir. Bu işlem genellikle toplama işlemi ile gerçekleştirilir ancak çarpımsal, signum, kümülatif toplam, maksimum ve minimum gibi yaklaşımlar da kullanılmaktadır (Haykin, 2009, s.40).

2.5.4 Girdi Katmanı - Çıktı Katmanı - Gizli Katman

Yapay sinir ağının ilk katmanı ve ağın girişi olan katmanı girdi katmanı, ağın çıkışını üreten ve ağın son katmanı ise çıktı katmanıdır. Süreç katmanları olarak bilinen gizli katmanlar, gelen veriler üzerindeki belirli görevleri uygular ve bu görevler sonucunda oluşturulan çıktıyı bir sonraki katmana aktarırlar. Çok katmanlı bir ileri beslemeli yapay sinir ağı Şekil 4'te verilmiştir.



Şekil 4: Çok Katmanlı Bir İleri Beslemeli Yapay Sinir Ağı

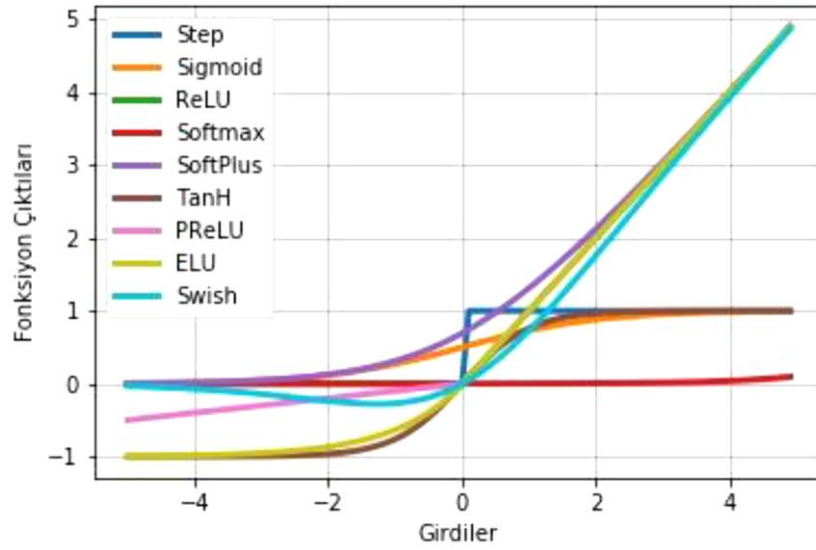
Kaynak: Mutlu Avcılar ve Emre Yakut, “Yapay Sinir Ağları Çoklu Lojistik Regresyon ve Çoklu Diskriminant Analiz Yöntemlerinden Yararlanarak Yerel Seçimlerde Seçmen Tercihlerinin Belirlenmesi: Osmaniye ili Uygulaması”, *Alanya İşletme Fakültesi Dergisi*, 7(2), 2015, s.219

Şekil 4’te de görüldüğü gibi, karmaşık problemlerde tek nöronun yeterli olmadığı durumlarda nöron yığınları kullanılmaktadır. Temelde basit bir ağı; giriş katmanı, bir gizli katman ve çıkış katmanından oluşmaktadır.

2.5.5 Aktivasyon Fonksiyonları

Biyolojik nöronda çekirdekteki bilginin işlenip çıktıya dönüştürülmesi işlemi, yapay nöronda toplam fonksiyonun oluşturduğu net girdinin aktivasyon fonksiyonundan geçirilerek çıktıya dönüştürülmesi ile gerçekleştirilir. Aktivasyon fonksiyonu, çıktının genişliğini belirli bir aralığa sıkıştıran ya da bir sınırlandırma getiren “sıkıştırma fonksiyonu” olarak da ifade edilebilen bir fonksiyondur (Haykin, 2009, s.32). Net girdi $u = \sum X_i w_i + b$ olmak üzere aktivasyon fonksiyonu uygulanması ile $\hat{y} = f(u)$ çıktısı elde edilir.

Problemin türü ve ağı yapısına göre aktivasyon fonksiyonun seçiminde farklı kullanımlara ihtiyaç duyulmaktadır. En yaygın olarak kullanılan aktivasyon fonksiyonlarına ilişkin performans karşılaştırılması yapılmıştır.



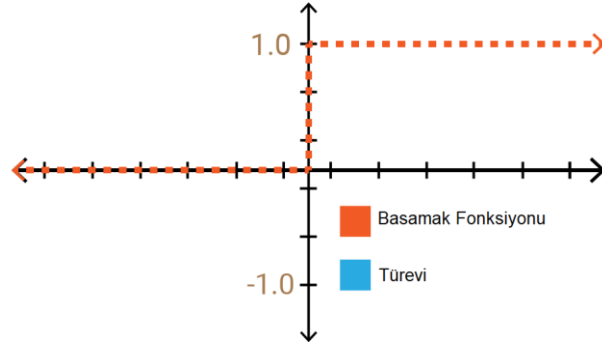
Şekil 5: Aktivasyon Fonksiyonlarının Performans Karşılaştırılması

Kaynak: Betül Ay Karakuş, “Derin Sinir Ağları için Aktivasyon Fonksiyonları”, *Büyük Veri ve Yapay Zeka Laboratuvarı*, 2018 <http://buyukveri.firat.edu.tr/2018/04/17/derin-sinir-aglari-icin-aktivasyon-fonksiyonlari/> (05 Mayıs 2021).

Çalışmalarda yaygın olarak kullanılan aktivasyon fonksiyonları aşağıdaki kısımda açıklanmıştır.

- **Birim Basamak (Step) Aktivasyon Fonksiyonu**

0 ve 1 değerlerini alan bir fonksiyon olması sebebiyle ikili sınıflayıcı olarak kullanılmaktadır. Eğer girdi negatifse 0, diğer durumlarda ise 1 değerini almaktadır. İkili değer alan bir fonksiyon olması sebebiyle genellikle çıkış katmanlarında tercih edilir. Bu fonksiyonun gizli katmanlarda kullanılması durumunda sadece 0 ve 1 değerleri üretileceğinden diğer katmanlara geçiş mümkün olmayacaktır. Ağırlıkların güncellenmesinde türev kullanımına ihtiyaç duyulacağından ancak fonksiyonun yapısı gereği türevi, öğrenme işine katkı sağlayan bir değer oluşturamadığı için kullanılması tavsiye edilmemektedir. Birden fazla gizli katman bulunması sebebiyle de derin öğrenme algoritmalarında kullanılmamakta, genellikle tek nöronlu modeller için kullanılmaktadır (Mehrotra ve diğerleri, 1996, s.11).

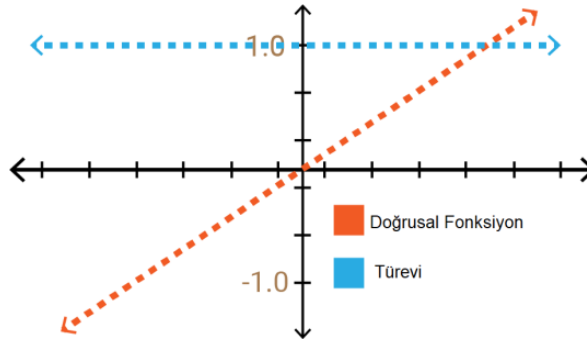


Şekil 6: Basamak Fonksiyonu ve Türevi

Kaynak: Ayyüce Kızrak, “Derin Öğrenme İçin Aktivasyon Fonksiyonlarının Karşılaştırılması”, *Medium*, 2019, <https://ayyucekizrak.medium.com/derin-%C3%B6%C4%9Frenme-i%C3%A7in-aktivasyon-fonksiyonlar%C4%B1n%C4%B1n-kar%C5%9F%C4%B1la%C5%9Ft%C4%B1r%C4%B1lmas%C4%B1-cee17fd1d9cd> (13 Nisan 2021).

- **Lineer (Doğrusal) Aktivasyon Fonksiyonu**

Bağımlı değişkenin bağımsız değişkene bağlı olarak doğrusal bir yapıda değişim gösterdiği fonksiyonlardır. Öğrenme işlemi için kullanılacak bir türev değeri elde edilmekle birlikte, hep aynı doğrusal yapının aktarılması öğrenme işleminin doğru bir şekilde gerçekleştiği anlamına gelmemektedir. Yani giriş katmanı ile çıkış katmanı arasında hep aynı doğrusal sonucun olması aradaki gizli katmanların işlevini ortadan kaldıracak için sığ mimarilerde kullanımı uygun bir fonksiyondur.



Şekil 7: Doğrusal Fonksiyon ve Türevi

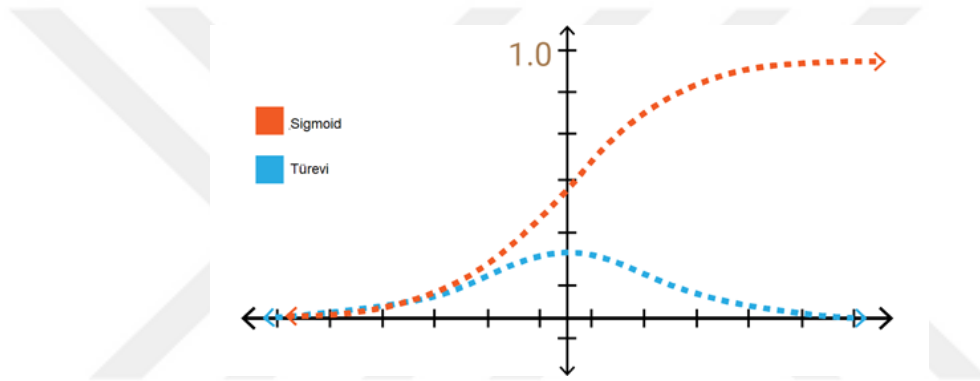
Kaynak: Ayyüce Kızrak, “Derin Öğrenme İçin Aktivasyon Fonksiyonlarının Karşılaştırılması”, *Medium*, 2019, <https://ayyucekizrak.medium.com/derin-%C3%B6%C4%9Frenme-i%C3%A7in-aktivasyon-fonksiyonlar%C4%B1n%C4%B1n-kar%C5%9F%C4%B1la%C5%9Ft%C4%B1r%C4%B1lmas%C4%B1-cee17fd1d9cd> (13 Nisan 2021).

- **Sigmoid (Lojistik) Aktivasyon Fonksiyonu**

En yaygın kullanılan aktivasyon fonksiyonudur. S tipi fonksiyon olarak da bilinmektedir. Değerleri (0-1) aralığına taşımaktadır ve

$$f(u) = \frac{1}{1 + e^{-u}}$$

olarak ifade edilmektedir. $(-\infty, \infty)$ aralığını olasılık değerlerine dönüştürmektedir. Lojistik fonksiyonun bir türü olan sigmoid aktivasyon fonksiyonu kategorik değişkenlerin varlığı durumunda tercih edilmektedir.



Şekil 8: Sigmoid Fonksiyonu ve Türevi

Kaynak: Ayyüce Kızrak, “Derin Öğrenme İçin Aktivasyon Fonksiyonlarının Karşılaştırılması”, *Medium*, 2019, <https://ayyucekizrak.medium.com/derin-%C3%B6%C4%9Frenme-i%C3%A7in-aktivasyon-fonksiyonlar%C4%B1n%C4%B1n-kar%C5%9F%C4%B1la%C5%9Ft%C4%B1r%C4%B1lmas%C4%B1-ccc17fd1d9cd> (13 Nisan 2021).

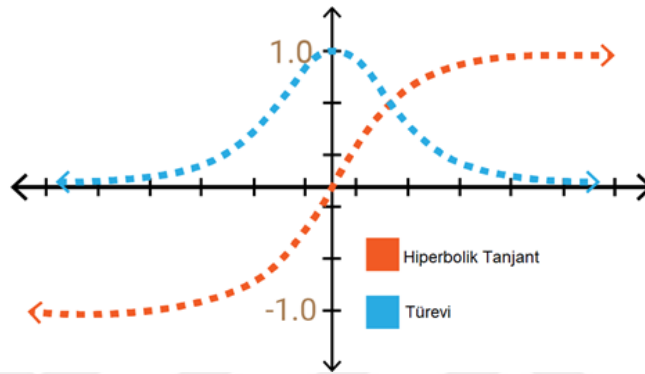
İkili sınıflama problemlerinde çıktı katmanında kullanılmaktadır. Katmanlardan elde edilen çıkış değeri 0 ve 1 değerlerine yaklaştıkça, modelin çıktısının parametrelere göre gradyanı 0 olacaktır. Bu durum, öğrenme algoritmasının parametrelerini güncellemesinde problem oluşturmaktadır (Goodfellow ve diğerleri, 2016, s.67).

- **Hiperbolik Tanjant (Tanh) Aktivasyon Fonksiyonu**

Sigmoid fonksiyonuna göre fonksiyon $(-1,1)$ aralığında değerler almaktadır ve

$$f(u) = \frac{e^u - e^{-u}}{e^u + e^{-u}}$$

olarak hesaplanmaktadır. Hiperbolik tanjant fonksiyonunun türevinin sigmoid fonksiyonuna göre daha dik değerler alması ve gradyan değerlerinde daha çok değer alabilmesi avantaj sağlamaktadır. Öğrenme hızını artırması ve sınıflama işlemi için daha geniş aralığa sahip olmasından dolayı daha yaygın olarak kullanılmaktadır. Ancak benzer şekilde fonksiyonun çok küçük ve çok büyük değerlerinde gradyanların hesaplanma problemi vardır (Heaton, 2015, s.44).



Şekil 9: Tanh Fonksiyonu ve Türevi

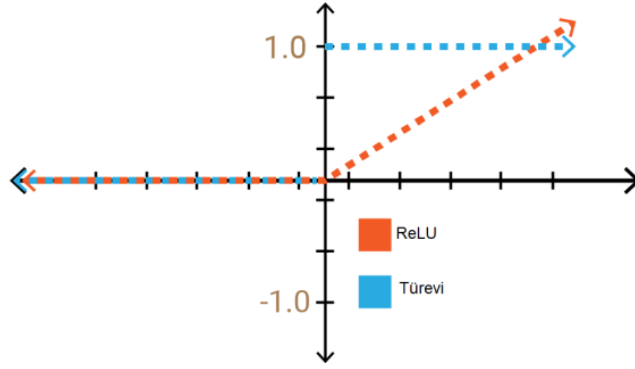
Kaynak: Ayyüce Kızrak, “Derin Öğrenme İçin Aktivasyon Fonksiyonlarının Karşılaştırılması”, *Medium*, 2019, <https://ayyucekizrak.medium.com/derin-%C3%B6%C4%9Frenme-i%C3%A7in-aktivasyon-fonksiyonlar%C4%B1n%C4%B1n-kar%C5%9F%C4%B1la%C5%9Ft%C4%B1r%C4%B1lmas%C4%B1-cee17fd1d9cd> (13 Nisan 2021).

- **Softmax Aktivasyon Fonksiyonu**

Genellikle sınıflandırma problemleri için çıktı katmanında kullanılmaktadır. Çok katmanlı sınıflandırma problemlerinde her bir sınıfa olasılıkların kolay yorumlanmasına olanak veren değerlerin atanmasını kolaylaştırıldığından tercih edilmektedir (Yılmaz ve Kaya, 2019). Bu fonksiyon ikiden fazla sınıflandırma yapılması gereken durumlarda kullanılmaktadır ve bu anlamda sigmoid fonksiyonunun genellenmiş hali olarak düşünülmektedir (Goodfellow ve diğerleri, 2016, s.184).

- **ReLU (Rectified Linear Unit) Aktivasyon Fonksiyonu**

Bu aktivasyon fonksiyonu $f(x) = \max(0, x)$ olarak tanımlanmaktadır. Burada $X > 0$ için fonksiyon değeri girdi değeri olan X olmakta ve $X \leq 0$ için sonuç 0 olarak alınmaktadır. Bu fonksiyonun en büyük avantajı tüm pozitif girdiler için ağın daha hızlı eğitilmesini sağlayan sabit türev değerine sahip olmasıdır (Yılmaz ve Kaya, 2019).



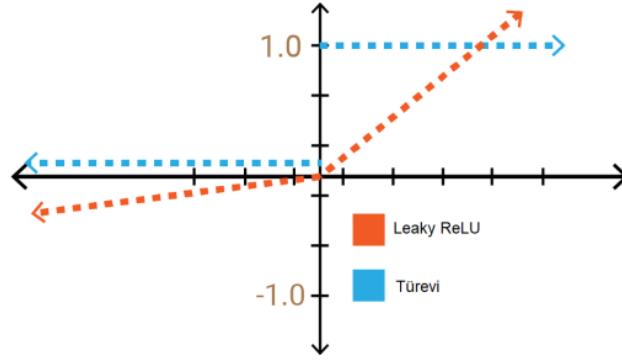
Şekil 10: ReLU Fonksiyonu ve Türevi

Kaynak: Ayyüce Kızrak, “Derin Öğrenme İçin Aktivasyon Fonksiyonlarının Karşılaştırılması”, *Medium*, 2019, <https://ayyucekizrak.medium.com/derin-%C3%B6%C4%9Frenme-i%C3%A7in-aktivasyon-fonksiyonlar%C4%B1n%C4%B1n-kar%C5%9F%C4%B1la%C5%9Ft%C4%B1r%C4%B1lmas%C4%B1-cee17fd1d9cd> (13 Nisan 2021).

ReLU aktivasyon fonksiyonu negatif değerlerde gradyan kaybına yol açmaktadır. Bu ortaya çıkan gradyan kaybı ile ağıın eğitilmesi sorunu ortaya çıkmaktadır. Ancak hesaplama yükünün sigmoid ve hiperbolik tanjant fonksiyonlarına göre az olması, çok katmanlı ağlarda daha çok tercih edilmesine sebep olmuştur (Sharma, 2017; Kızrak, 2019).

- **Sızıntı (Leaky) ReLU Aktivasyon Fonksiyonu**

ReLU fonksiyonunda $X \leq 0$ değerleri için sıfır değerini alıyor olması problemini ortadan kaldırmak adına Sızıntı (Leaky) ReLU Aktivasyon Fonksiyonu kullanılmaktadır. Başka bir ifade ile ReLU ile karşılaştırıldığında, sızıntı ReLU, negatif değerler için sıfır olmayan bir gradyan değeri vermekte ve gradyan kaybı önlenmektedir. Dolayısıyla negatif bölgede yer alan değer için de öğrenme sağlanmış olmaktadır (Xu, Huang ve Li, 2016).



Şekil 11: Sızıntı ReLU Fonksiyonu ve Türevi

Kaynak: Ayyüce Kızrak, “Derin Öğrenme İçin Aktivasyon Fonksiyonlarının Karşılaştırılması”, *Medium*, 2019, <https://ayyucekizrak.medium.com/derin-%C3%B6%C4%9Frenme-i%C3%A7in-aktivasyon-fonksiyonlar%C4%B1n%C4%B1n-kar%C5%9F%C4%B1la%C5%9Ft%C4%B1r%C4%B1lmas%C4%B1-cee17fd1d9cd> (13 Nisan 2021).

Sızıntı ReLU aktivasyon fonksiyonun genel hali,

$$f(x) = \max(\alpha x, x)$$

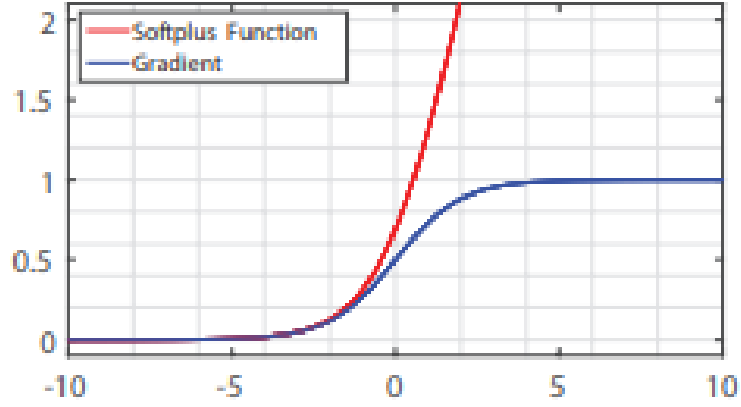
olarak tanımlanmaktadır. Burada sızıntı değeri olan α değeri, 0.01 gibi küçük bir değer olarak sabitlenir. Bu fonksiyonda α 'nın öğrenilebilir bir parametreymiş gibi keyfi değerler alması durumunda ise fonksiyon Parametrik ReLU (PReLU) aktivasyon fonksiyonu adını almaktadır (Goodfellow ve diğerleri, 2016, s.193).

- **Softplus Aktivasyon Fonksiyonu**

Softplus aktivasyon fonksiyonu ReLU fonksiyonuna benzemekle birlikte daha pürüzsüz bir düzgünleştirme sağlamaktadır. Softplus fonksiyonu,

$$f(x) = \log(1 + e^x)$$

şeklinde tanımlanmaktadır. Ancak softplus kullanımı her yerde türevlenebildiği için daha avantajlı görünse de deneysel olarak doğru olmadığı görüldüğünden genellikle kullanımı önerilmemektedir (Goodfellow ve diğerleri, 2016, s.68).



Şekil 12: Softplus Fonksiyonu ve Türevi

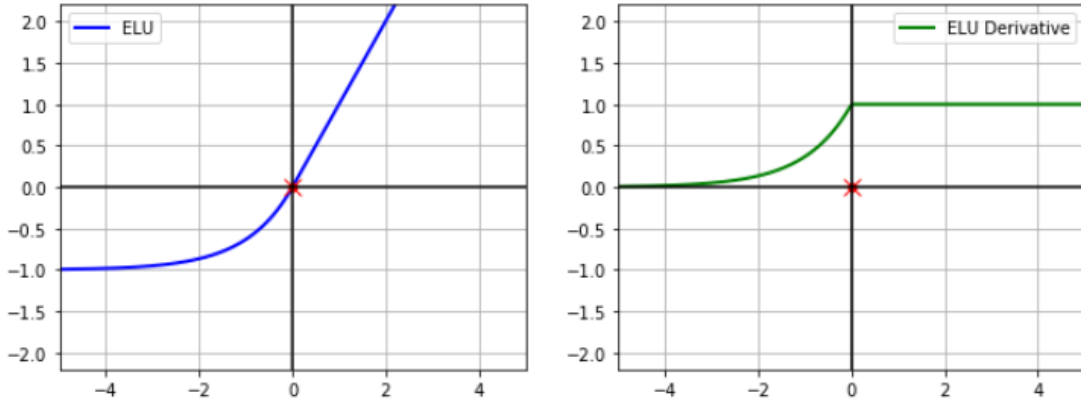
Kaynak: Giorgio Roffo, “Ranking to Learn and Learning to Rank: On The Role of Ranking in Pattern Recognition Applications”, *arXiv preprint arXiv:1706.05933*, 2017.

- **ELU Aktivasyon Fonksiyonu**

Üstel Doğrusal Birimler (ELU) Aktivasyon fonksiyonu,

$$f(x) = \begin{cases} x & , \quad x > 0 \\ \alpha(e^x - 1) & , \quad x \leq 0 \end{cases}$$

olarak tanımlanmaktadır. Fonksiyonda yer alan α hiperparametresi negatif girdi değerlerinin kontrolünü sağlamaktadır. Negatif değerlerde sıfır olmayan bir gradyana sahip olması sebebiyle, gradyan kaybı problemine karşı önlem oluşturmaktadır. ELU aktivasyon fonksiyonu, negatif olmayan girdilerin olduğu durumda ReLU ile çok benzer yapıdadırlar. Ancak ELU aktivasyon fonksiyonu, aktivasyonların ortalamasını sıfıra yaklaştıran negatif değerlere sahiptir. ELU’nun en büyük dezavantajı, ReLU’daki keskin düzgünleştirmenin aksine, üstel yapı sergilemesinden dolayı, daha yavaş bir şekilde düzgünleştirme yapmasıdır (Clevert, Unterthiner ve Hochreiter, 2016).

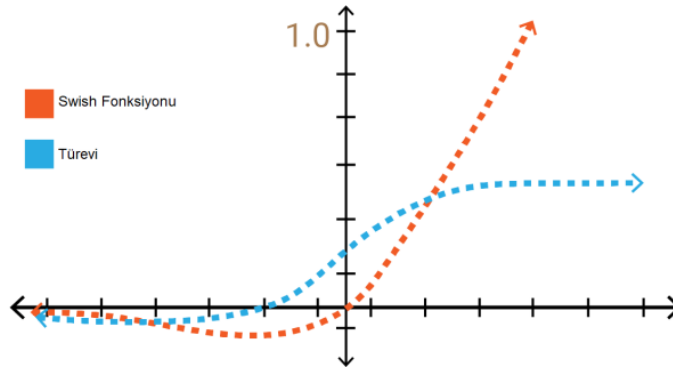


Şekil 13: ELU Fonksiyonu ve Türevi ($\alpha=1$ için)

Kaynak: Kshitij Khurana, “Activation Functions in Neural Networks”, *Medium*, 2019, <https://medium.com/@kshitijkhurana3010/activation-functions-in-neural-networks-ed88c56b611b> (12 Nisan 2021).

- **Swish (A Self-Gated/Kendinden Geçitli) Aktivasyon Fonksiyonu**

Swish aktivasyon fonksiyonu, negatif bölgede değer alan ve doğrusal olmayan bir fonksiyondur. Negatif bölgede değer alabilme özelliği ile ReLU aktivasyon fonksiyonundan ayrılırken, doğrusal olmama özelliği ile Leaky ReLU’dan ayrılmaktadır.



Şekil 14: Swish ReLU Fonksiyonu ve Türevi

Kaynak: Ayyüce Kızrak, “Derin Öğrenme İçin Aktivasyon Fonksiyonlarının Karşılaştırılması”, *Medium*, 2019, <https://ayyucekizrak.medium.com/derin-%C3%B6%C4%9Frenme-i%C3%A7in-aktivasyon-fonksiyonlar%C4%B1n%C4%B1n-kar%C5%9F%C4%B1la%C5%9Ft%C4%B1r%C4%B1lmas%C4%B1-ccc17fd1d9cd> (13 Nisan 2021).

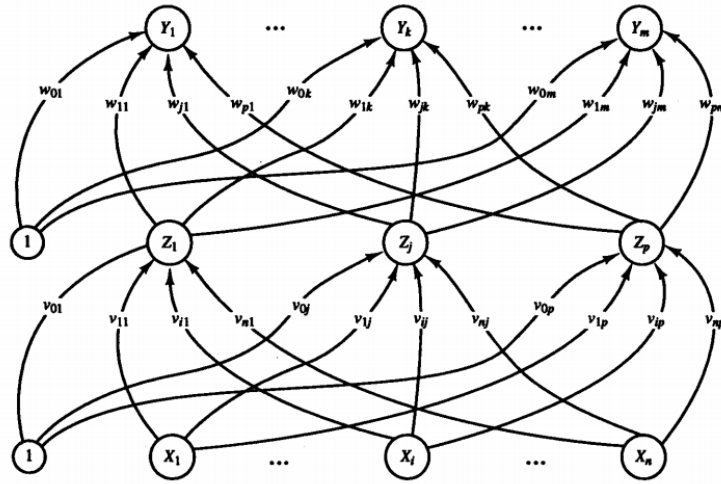
Swish aktivasyon fonksiyonu;

$$f(x) = 2x\sigma(\beta x)$$

olarak formülize edilmektedir. Burada yer alan $\sigma(\beta x)$ sigmoid fonksiyonudur. β sabit ya da eğitilebilir bir parametre olmak üzere; $\beta = 0$ ise fonksiyon doğrusal hale gelmektedir. $\beta \rightarrow \infty$ yani oldukça büyük bir değer olarak seçilirse sigmoid kısmı 0-1 ikili basamak fonksiyonuna dönüşeceğiinden ReLU fonksiyonuna benzeyecektir. Bu durumlar göz önünde bulundurularak standart Swish fonksiyonu için uygun değer $\beta = 1$ 'dir (Ramachandran, Zoph ve Le, 2017).

2.5.6 İleri Besleme ve Geri Yayılım

Sistemin ilerleme yönü dikkate alındığında, yapay sinir ağları iki grupta incelenebilir. Ara sinirlerden geçerek, giriş katmanından çıkış katmanına doğru sistemin tek yönlü olarak ilerlediği ağlara ileri beslemeli (feedforward) ağ denirken, herhangi bir sinirin çıkışından girişine dönüşün olduğu, yani geri dönüşlerin var olduğu ağlar geri beslemeli (feedback) ağlar olarak isimlendirilir (Nabiyev, 2021, s.590).



Şekil 15: Bir Gizli Katmanlı Geri Yayılım Algoritması

Geri yayılım algoritması temel olarak 3 aşamada gerçekleşir. İlk aşama *ileri besleme aşamasıdır*. Bu adımda girdiler ağıdaki katmanlardan adım adım çıktı aktivasyonları oluşturularak ilerletilir. İkinci aşama *geri beslemedir*. Burada maliyet fonksiyonları alınarak her bir katmandaki tartılara ilişkin kısmi türevler alınmaktadır. Son aşama ise *tartıların güncellenmesidir*. Tartılar ise çeşitli optimizasyon algoritmalarına bağlı olarak güncellenmektedir (Gürsakal, 2017, s.88). Geri yayılım algoritmasının detaylı algoritma adımları aşağıdaki gibidir (Fausett, 1993, s.294):

Adım 0: Başlangıç ağırlıkları küçük değerler olacak şekilde rassal olarak atanır.

Adım 1: Durdurma kuralı sağlanmadığı sürece Adım 2-9 gerçekleştirilir.

Adım 2: Her eğitim çifti için Adım 3-8 uygulanır.

İleri Besleme:

Adım 3: Her $X_i, (i = 1, \dots, n)$ girdi birimi için, x_i girdi sinyali alınır ve bir üst katmanda yer alan tüm gizli birimlere aktarılır.

Adım 4: Her $Z_j, (j = 1, \dots, p)$ gizli birimi için, ağırlıklı girdi sinyalleri toplanır.

$$z_in_j = v_{0j} + \sum_{i=1}^n x_i v_{ij}$$

Aktivasyon fonksiyonu kullanılarak çıktı sinyali elde edilir:

$$z_j = f(z_in_j)$$

Hesaplanan çıktı sinyali bir üst katmandaki tüm çıktı birimlerine gönderilir.

Adım 5: Her $Y_k, (k = 1, \dots, m)$ çıktı birimi için, ağırlıklı girdi sinyalleri toplanır.

$$y_in_k = w_{0k} + \sum_{j=1}^p z_j w_{jk}$$

Aktivasyon fonksiyonu kullanılarak çıktı sinyali elde edilir:

$$y_k = f(y_in_k)$$

Hatanın Geri Yayılımı:

Adım 6: Her $Y_k, (k = 1, \dots, m)$ çıktı birimi için, girdi eğitim örüntüsüne karşılık gelen bir hedef örüntü alınır ve hata bilgi terimi hesaplanır:

$$\delta_k = (t_k - y_k) f'(y_in_k)$$

α , öğrenme hızı olmak üzere; güncel w_{jk} ağırlıkları hesaplanırken kullanılacak ağırlık düzeltme terimi hesaplanır.

$$\Delta w_{jk} = \alpha \delta_k z_j$$

Güncel w_{0k} ağırlıkları hesaplanırken kullanılacak sapma düzeltme terimi hesaplanır:

$$\Delta w_{0k} = \alpha \delta_k$$

ve δ_k alt katmandaki birimlere gönderilir.

Adım 7: Her Z_j , ($j = 1, \dots, p$) gizli birimi için, üst katmandaki birimlerden gelen delta girdileri toplanır:

$$\delta_{in_j} = \sum_{k=1}^m \delta_k w_{jk}$$

hata bilgi teriminin hesaplanması için aktivasyon fonksiyonun türevi ile çarpma işlemi yapılır.

$$\delta_j = \delta_{in_j} f'(z_{in_j})$$

Güncel v_{ij} ağırlıkların hesaplanmasında kullanılacak ağırlık düzeltme terimi hesaplanır:

$$\Delta v_{ij} = \alpha \delta_j x_i$$

Güncel v_{0j} ağırlıklarının hesaplanmasında kullanılmak üzere sapma düzeltme terimi hesaplanır.

$$\Delta v_{0j} = \alpha \delta_j$$

Ağırlıkları ve sapmaları güncellemek:

Adım 8: Her Y_k , ($k = 1, \dots, m$) çıktı birimi için sapma ve ağırlıklar güncellenir.

$$w_{jk}(yeni) = w_{jk}(eski) + \Delta w_{jk}$$

Her Z_j , ($j = 1, \dots, p$) gizli birimi için sapma ve ağırlıklar güncellenir:

$$v_{ij}(yeni) = v_{ij}(eski) + \Delta v_{ij}$$

Adım 9: Durdurma kuralı test edilir.

2.5.7 Kayıp - Maliyet Fonksiyonu

Ağ düzenlerinin amacı, gerçek değerlere en yakın çıktıların tahmin edilmesidir. Bir ağın doğruluk değeri, maliyet ya da kayıp fonksiyonu kullanılarak hesaplanmaktadır. Tahmin doğruluğunu artırarak hatanın azaltılması maliyet fonksiyonunun minimize edilmesi problemidir. Burada, hata hesaplama problemi optimizasyon problemine dönüşmektedir. Bu sebeple, kayıp fonksiyonu (Loss function) kavramı amaç fonksiyonu, hata fonksiyonu veya maliyet fonksiyonu (Cost function) olarak da ifade edilmektedir. Bu kavramlar birbirinin yerine kullanılabilirken, bazı kaynaklara göre bu kavramlara özel bir anlam yüklenmektedir (Goodfellow ve diğerleri, 2016, s.82). Kayıp fonksiyonu temelde, eğitim sonucunda elde edilen ağırlık değeri (w) ve sapma (b) parametrelerinin problem çözümü için ne kadar uygun olduğunu hesaplamaktadır (Alammar, 2016). m eğitilen girdilerin sayısı, y gerçek değer ve $\hat{y}$ tahmin değeri olmak üzere maliyet fonksiyonu;

$$L = \frac{1}{m} \sum_{i=1}^m (y_i - \hat{y}_i)^2$$

ortalama hata kareler ifadesi ile elde edilebilir. Bu formülde gösterildiği gibi, birden fazla girdi değeri için kayıp hesaplamak maliyet fonksiyonunu oluşturacaktır. Aynı formül tek girdi için hesaplandığında kayıp değerini oluşturacaktır (Mulla, 2020).

Maliyeti minimize etmek için gradyan iniş algoritması kullanılmaktadır. Minimum maliyet noktasının bulunması amacıyla, fonksiyonun yerel minimumu bulmak için fonksiyonun negatif eğimine göre ilerlenmelidir. Ayrıca maliyet fonksiyonunu minimize etmek için her iterasyonda (yineleme sayısı) elde edilen en aza indirgeme

miktarı öğrenme oranı olarak ifade edilmektedir. En uygun sonucun elde edilmesi için bu oranın seçimi çok önemlidir. Aksi halde en uygun çözüm elde edilmeyebilir ya da ağın en uygun çözüme yaklaşması sonsuz olabilir (Goodfellow ve diğerleri, 2016, s.83). Maliyet fonksiyonunun hesaplanmasında kullanılan hata ölçütleri regresyon ve sınıflandırma olmak üzere iki ana grup altında toplanmaktadır. Ayrıca derin öğrenme ile nesne ve yüz tanımlama için kullanılan kayıp fonksiyonlarının detaylı listesi Tablo 1’de yer almaktadır (Wang ve diğerleri, 2020).

Tablo 1
Kayıp Fonksiyonları

Kategori	Uygulama	Kayıp Fonksiyonu
Geleneksel Makine Öğrenmesi	Sınıflandırma	0-1 kaybı, Algılayıcı kaybı, Logaritmik kayıp, Üstel kayıp, Sigmoid çapraz entropi kaybı, Softmax çapraz entropi kaybı, Hinge kaybı, Ramp kaybı, Pinball kaybı,
	Regresyon	Karesel kayıp, Mutlak kayıp, Huber kaybı, log-cosh kaybı, Kantil kaybı, e-insensitive kaybı
	Denetimsiz öğrenme	Karesel hata, Uzaklık hatası, Yeniden yapılandırma hatası, Negatif varyans
Derin Öğrenme	Nesne tanıma	IoU kaybı, Genelleştirilmiş IoU kaybı, Düzgünleştirilmiş L1 kaybı, Odak kaybı
	Yüz tanıma	Karşılaştırmalı kayıp, Üçlü kayıp, Merkez kaybı, Açısız softmax kaybı (A-Softmax Kaybı), Toplamsal marj softmax kaybı (AM-Softmax kaybı), Toplamsal açısız marj kaybı (ArcFace kaybı)

Temel teknikler olması sebebiyle regresyon ve sınıflandırma problemleri için yaygın olarak kullanılan kayıp fonksiyonlar detaylı olarak açıklanmıştır.

2.5.7.1 Sınıflandırma İçin Kullanılan Kayıp Fonksiyonları

- **0 - 1 Kayıp Fonksiyonu:**

0-1 kayıp fonksiyonunda tahmin edilen değer ($f(x)$) ile gerçek değer (y) olmak üzere;

$$L(y, f(x)) = \begin{cases} 1 & , \quad yf(x) < 0 \\ 0 & , \quad yf(x) \geq 0 \end{cases}$$

şeklinde tanımlanmaktadır. Tahmin değeri ile gerçek değer aynı işarete sahipse kayıp değeri 0; aksi halde kayıp değeri 1 olacaktır. Bu hesaplamada, tahmin edilen değer ile gerçek değer arasındaki hata derecesi dikkate alınmamaktadır. Yani örnekler yanlış sınıflandırıldığında, ne kadar hata olursa olsun, kayıp değerleri aynı olacaktır. Dolayısıyla

modelin doğruluğu ve verimliliği etkilenecektir. Ayrıca 0-1 kayıp fonksiyonunun grafiğinden dışbükey olmayan bir fonksiyon olduğu ve modelin çözüm sürecinde birçok eksiklik olduğu görülmektedir (Wang ve diğerleri, 2020).

- **Algılayıcı (Perceptron) Kaybı Fonksiyonu:**

Algılayıcı kaybı fonksiyonu parçalı bir fonksiyondur ve

$$L(y, f(x)) = \max\{0, -yf(x)\}$$

olarak hesaplanmaktadır. Ayrıca,

$$\max\{a, b\} = \begin{cases} a & , a \geq b \\ b & , a < b \end{cases}$$

şeklinde ifade edilmektedir. Tahmin edilen değer gerçek değer ile aynı işarete sahip olduğunda kayıp değeri 0'dır; aksi takdirde kayıp değeri, tahmin edilen değerın mutlak değeridir. Kaybın sıfır olması doğru sınıflandırılmış örnekler için kayıp olmadığı anlamına gelirken, ikinci durumda tahmin edilen örneklerden karar sınırına olan uzaklık ne kadar büyükse, hata o kadar büyük olur. Algılayıcı kaybında kayıp fonksiyonu sürekli ve değişkenlere göre türevlenebilir, böylece optimize edilmesi kolaydır. Amacı sadece örneklerin kategorisini doğru belirlemektir. Bu şekilde elde edilen modelin genelleme performansı zayıf ve gürültü verilerine karşı sağlam değildir (Wang ve diğerleri, 2020).

- **Logaritmik Kayıp Fonksiyonu:**

Logaritmik kayıp fonksiyonu, “çapraz entropi” veya “lojistik kayıp” olarak da anılmaktadır (Brownlee, 2019). Bu kayıp fonksiyonu, tahmini olasılık değerinin koşullu olasılık dağılımı ile elde edildiği bir fonksiyondur. Olasılıkların tahmini, 0 ile 1 arasında sayıların üretilmesi anlamına gelmektedir. Olasılıklarla çalışma söz konusu olduğunda logaritmik dönüşümler kullanılmaktadır (Patterson ve Gibson, 2017, s.144). Örnek etiketinin tahmin edilme olasılığı ne kadar yüksek olursa, buna karşılık gelen kayıp değeri o kadar küçük olur. Gerçek hesaplamada pozitif sınıf olarak tahmin edilme olasılığı p olarak ifade edilirken, diğer olasılık $(1 - p)$ 'dir. Kayıp fonksiyonu;

$$L(y, \tilde{p}) = -\log \tilde{p}$$

olarak ifade edilmektedir. Burada $\tilde{p}$;

$$\tilde{p} = \begin{cases} p & , y = 1 \\ 1 - p & , y \neq 1 \end{cases}$$

olup,

$$p = P(y = 1|x) = \frac{1}{1+e^{-f(x)}} \quad \text{ve} \quad 1 - p = P(y = -1|x) = \frac{1}{1+e^{f(x)}}$$

şeklinde gösterilmektedir. Doğru tahmin olasılığının 1'e yakın olması durumunda logaritmik kayıp değerinin yavaş yavaş azaldığı ve doğru tahmin olasılığının azalmasıyla kayıp değerinin hızla arttığı görülmektedir (Wang ve diğerleri, 2020).

- **Sigmoid Çapraz Entropi Kayıp Fonksiyonu:**

Sigmoid çapraz entropi kaybının formülü ile logaritmik kayıp formülü aynıdır. Ancak sigmoid çapraz entropi kaybı, sigmoid aktivasyon fonksiyonu aracılığıyla elde edilmektedir. Çapraz entropi, gerçek çıktı ile beklenen çıktı arasındaki uzaklığı tanımlamaktadır. Kayıp fonksiyonu;

$$L(y, \tilde{p}) = -\log \tilde{p}$$

olarak ifade edilmektedir. Burada $\tilde{p}$;

$$\tilde{p} = \begin{cases} p & , y = 1 \\ 1 - p & , y \neq 1 \end{cases}$$

dir. Burada $\sigma(\cdot)$ Sigmoid fonksiyonu olmak üzere, $p = \sigma(f(x)) = \frac{1}{1+e^{-f(x)}}$ olarak ifade edilmektedir. Çapraz entropinin değeri ne kadar küçükse, iki olasılık dağılımı da o kadar yakın olur. Bu nedenle, sigmoid çapraz entropi kaybının anlamı logaritmik kayıpla aynıdır. Çıktı katmanındaki nöronların yavaş öğrenme oranını önlemek için sinir ağlarında sigmoid çapraz entropi kaybı kullanılmaktadır (Kohonen ve diğerleri, 1988).

- **Softmax Çapraz Entropi Kayıp Fonksiyonu:**

Bu kayıp fonksiyonu ile sigmoid çapraz entropi kaybı fonksiyonu arasındaki fark, bu kayıp fonksiyonunun çoklu sınıflandırma problemini çözmek için sigmoid fonksiyonunu softmax fonksiyonu ile değiştirmesidir. Softmax çapraz entropi kaybı,

$$L(y, P(y|x)) = -\log P(y|x)$$

olarak tanımlanmaktadır. Burada $f_y(x)$ ve $f_k(x)$, sırasıyla y ve k sınıfına karşılık gelen karar fonksiyonları olmak üzere $P(y|x) = \frac{e^{f_y(x)}}{\sum_k e^{f_k(x)}}$ olarak ifade edilmektedir.

Semantik segmentasyon, metin madenciliği gibi çoklu sınıflandırma problemleri olduğunda, softmax çapraz entropi kaybı, derin öğrenmede ana kayıp fonksiyonu haline gelmiştir. Derin öğrenme literatürünün çoğunda, hem sigmoid hem de softmax çapraz entropi kaybı, problemin kendisi tarafından belirlenebilen çapraz entropi kaybı olarak adlandırılır.

- **Üstel Kayıp Fonksiyonu:**

Bu kayıp fonksiyonu 0-1 kayıp fonksiyonu yaklaşımına benzerdir. Ancak 0-1 kaybının dezavantajlarını iyileştirmektedir. Sürekli bir fonksiyon olması sebebiyle türevlenebilirdir. Bu durum da optimizasyona olanak sağlamaktadır. Ayrıca yanlış sınıflandırılmış örnekler için hata derecelerine bağlı olarak üstel kaybın ceza dereceleri farklıdır ve doğru sınıflamanın da bir maliyeti vardır. Üstel kayıp fonksiyonu,

$$L(y, f(x)) = e^{-yf(x)}$$

olarak hesaplanmaktadır. Hata derecesi arttıkça, kayıp değeri daha hızlı artmaktadır. Bu durum modelin yakınsaması için elverişli bir yapı ortaya koymaktadır. Üstel kayıp fonksiyonunun temsili algoritması, 1995 yılında Yoav Freund ve Robert Schapire tarafından önerilen bir ikili sınıflandırma modeli olan AdaBoost'tur (Freund ve Schapire, 1995). Ancak gürültülü verilerde sorunlara neden olabilmektedir (Weinberger, 2015).

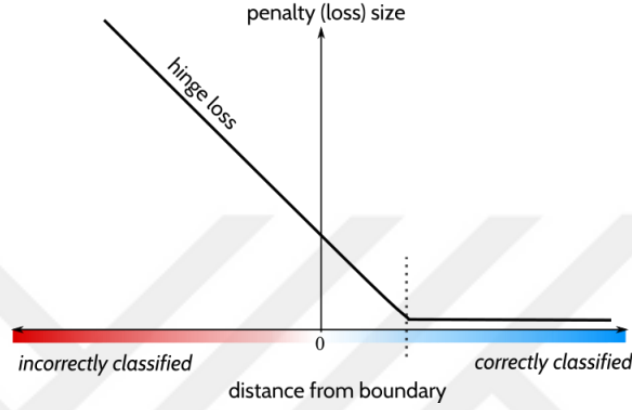
- **Hinge Kayıp Fonksiyonu:**

Hinge kaybı, klasik Destek Vektör Makineleri yöntemini temel almaktadır ve

$$H_s(z) = \max\{0, s - z\}$$

$$L(y, f(x)) = H_1(yf(x)) = \max\{0, 1 - yf(x)\}$$

olarak tanımlanmaktadır.

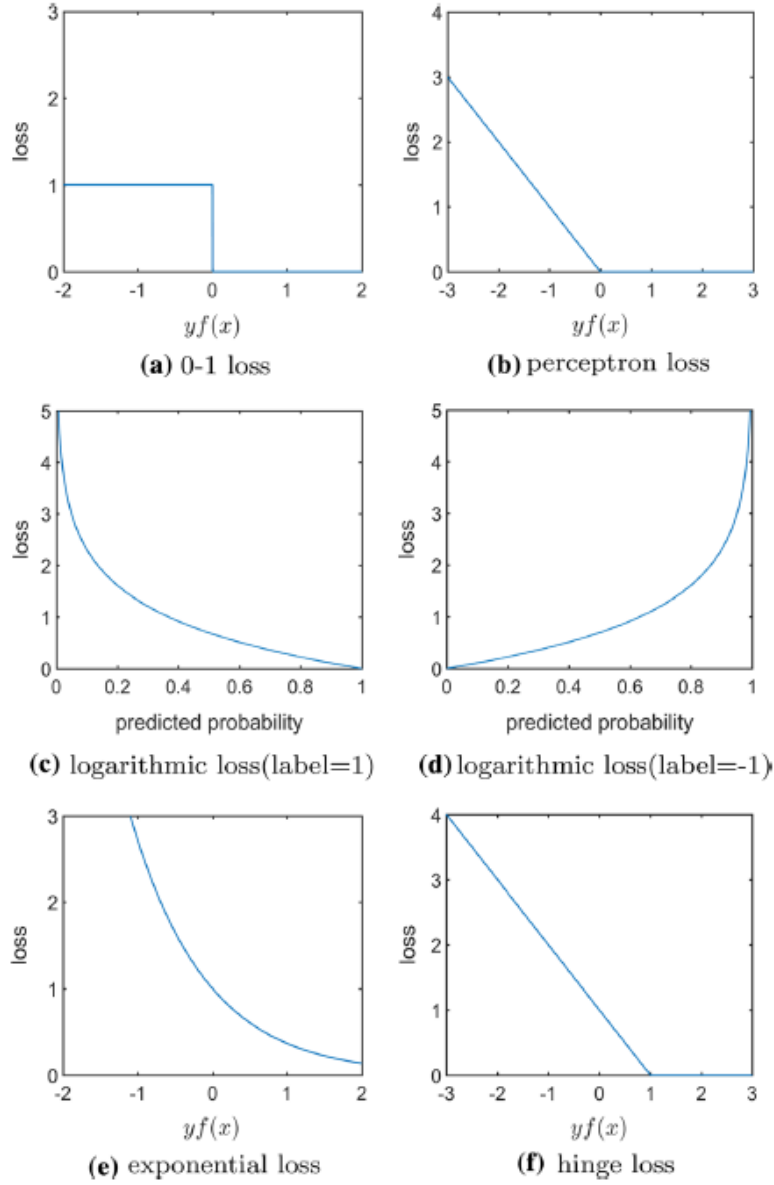


Şekil 16: Hinge Kayıp Fonksiyonu

İkili sınıflandırma probleminde, hinge kaybı fonksiyonu karar sınırına yakın s olarak ifade edilen bir eksen (marjin) tanımlar. Hinge kaybı fonksiyonunun uygulamalarında ve iyileştirmelerinde s genellikle 1 olarak ayarlanmaktadır. Karar sınırından uzaklık 1 veya 1'den büyük olduğunda kayıp fonksiyonu 0 olacaktır. Sınırdan uzaklık 0 ise o zaman kayıp fonksiyonu 1 olacaktır. Karar sınırından negatif uzaklıklar, sınırın yanlış tarafında olunmasından dolayı yanlış sınıflamaya sebep olacağından, yüksek kayıp oluşturacaktır. Sınırın ne kadar pozitif yönde uzağında kalınırsa hinge kaybı o kadar düşük olacaktır (Aliyev, 2020).

Ek olarak, üstel kayıp fonksiyonundan farklı olarak, hinge kaybı fonksiyonu, $|f(x)| \geq s$ olduğunda doğru sınıflandırma örneklerini cezalandırmaz. Modelin genel sınıflandırma hatasına daha fazla odaklanabilmesi için bu tür örneklerin yeterince iyi öğrenildiği düşünülmektedir. Ayrıca destek vektör makineleri (SVM'ler), ikili sınıflandırma problemini çözmek için hinge kaybını kullanan ilk algoritmadır (Wang ve diğerleri, 2020).

Şekil 17'de sınıflandırma için anlatılan tüm kayıp fonksiyonlarının toplu olarak şekilleri yer almaktadır.



Şekil 17: Sınıflandırma Kayıp Fonksiyonlarının Grafikleri

Kaynak: Qi Wang, Yue Ma, Kun Zhao ve Yingjie Tian, “A Comprehensive Survey of Loss Functions in Machine Learning”, *Annals of Data Science*, 9(2), s. 187-212, 2020.

2.5.7.2 Regresyon İçin Kullanılan Kayıp Fonksiyonları

- **Karesel Kayıp Fonksiyonu:**

Regresyon problemlerinde en yaygın olarak kullanılan kayıp fonksiyonlarından biridir. Gerçek değer ile tahmin değeri arasındaki uzaklığın karesel

olarak toplanması ile elde edilir. Ayrıca kuadratik kayıp veya L2 kaybı olarak isimlendirilmektedir ve gerçek değer y ve tahmin değeri $f(x)$ olmak üzere

$$L(y, f(x)) = (y - f(x))^2$$

olarak tanımlanmaktadır.

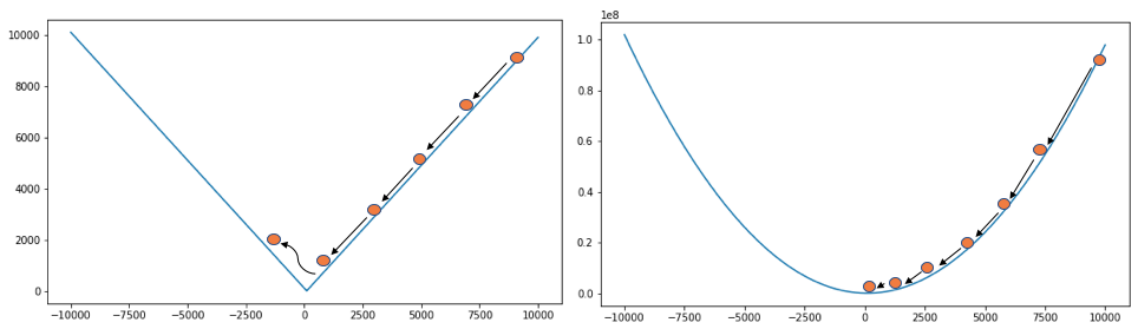
Şekil 18’den de görüleceği gibi karesel kaybın gradyanı her noktada değişkenlik göstermektedir. Hızlı yakınsama sağlaması ve doğruluğun yüksek olması avantaj sağlamaktadır. Dezavantajı aykırı değerlere karşı duyarlı olmasıdır. Sınıflandırma problemlerinde de kullanılmaktadır. Gerçek değer ile tahmin edilen sınıf olasılığı arasındaki farkı ölçmektedir (Wang ve diğerleri, 2020).

- **Mutlak Değer Kayıp Fonksiyonu:**

Regresyon problemlerinde kullanılan diğer bir kayıp fonksiyonudur. L1 kayıp fonksiyonu olarak da adlandırılmaktadır. Gerçek değer ile tahmin değeri arasındaki farkın mutlak değerini ölçer ve

$$L(y, f(x)) = |y - f(x)|$$

şeklinde ifade edilir.



Şekil 18: Karesel Kayıp ve Mutlak Kayıp Karşılaştırılması

Kaynak: Prince Grover, “Regression Loss Functions All Machine Learners Should Know”, 2018 <https://heartbeat.comet.ml/5-regression-loss-functions-all-machine-learners-should-know-4fb140e9d4b0> (11 Mayıs 2021).

Hatanın artması ile mutlak hata hızlı bir şekilde artmamaktadır. Bu sebeple aykırı değerlere karşı karesel kayba göre daha sağlamdır. Ancak gradyan iniş algoritmasındaki

ilerlemesi daha hızlı olduğundan 0'a yaklaştığında minimum noktayı atlama riski söz konusudur (Parmar, 2018; Grover, 2018).

- **Huber Kayıp Fonksiyonu:**

Huber kaybı, karesel kayıp ile mutlak kaybın avantajlarını etkin bir şekilde birleştirir. Huber kaybı, verilerdeki aykırı değerlere karesel hata kaybından daha az duyarlıdır. Ayrıca ve her yerde türevlenebilir olduğundan mutlak kayba göre avantaj sağlamaktadır. Huber kaybında $\delta > 0$ olmak üzere,

$$L_{\delta}(y, f(x)) = \begin{cases} \frac{1}{2}(y - f(x))^2 & , |y - f(x)| \leq \delta \\ |y - f(x)| - \frac{1}{2}\delta^2 & , \text{diğer durumlarda} \end{cases}$$

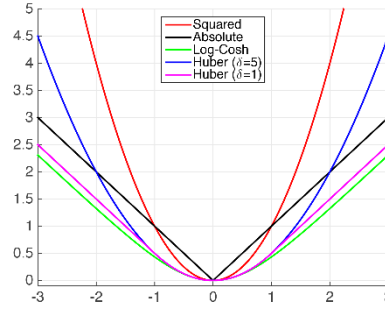
olarak ifade edilmektedir. δ değeri azaldığında (sıfıra yaklaştığında) karesel kayıp fonksiyonu (L2), δ değeri arttığında ise mutlak kayıp fonksiyonu (L1) gibi davranmaktadır. Huber kaybının özel bir şekli olan pürüzsüz mutlak hatalar ortalaması (smooth mean absolute error) derin öğrenmede de kullanılır (Grover, 2018; Wang ve diğerleri, 2020).

- **Log-cosh Kayıp Fonksiyonu:**

Bu kayıp fonksiyonu, hatanın hiperbolik kosinüsünün logaritmasıdır ve

$$L(y, f(x)) = \log(\cosh(f(x) - y))$$

olarak gösterilmektedir. Hata küçük olduğunda, kayıp yaklaşık $\frac{1}{2}(y - f(x))^2$ olacaktır. Hata büyük olduğunda ise, kayıp yaklaşık olarak $|y - f(x)| - \log 2$ olacaktır. Log-cosh kayıp fonksiyonu formülünden, Huber kaybına çok benzediği görülmektedir. Bu sebeple Huber kaybının tüm avantajlarına sahiptir, yani aykırı değerlere karşı sağlamdır ve her yerde türevlenebilirdir. Ek olarak, log-cosh kaybı her yerde ikinci dereceden türevlenebilir olması, XGBoost gibi birçok makine öğrenme algoritmasında, optimizasyon problemlerini çözmek için Newton yöntemini kullandığından avantaj sağlamaktadır (Weinberger, 2015; Grover, 2018; Wang ve diğerleri, 2020).



Şekil 19: Regresyon Kayıp Fonksiyonları

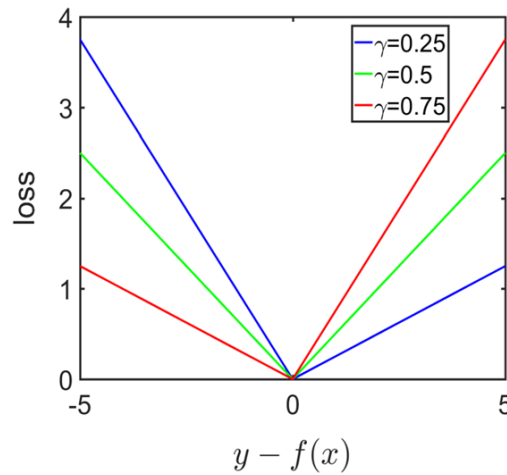
Kaynak: Kilian Weinberger, “CS4780/CS5780: Machine Learning for Intelligent Systems” 2015, <http://www.cs.cornell.edu/courses/cs4780/2015fa/web/lecturenotes/lecturenote10.html> (11 Mart 2022).

- **Kantil Kayıp Fonksiyonu:**

Bu kayıp fonksiyonu aslında mutlak kaybın genişletilmiş bir şeklidir ve 50. persantil alındığında mutlak kayba dönüşmektedir. Önceki kayıp fonksiyonlarından farklı olarak, nicel kayba dayalı regresyon modeli tahmin aralığı vermektedir. Kantil kayıp fonksiyonu $\gamma \in (0,1)$ olmak üzere,

$$L_{\gamma}(y, f(x)) = \begin{cases} (\gamma - 1)(y - f(x)), & y < f(x) \\ \gamma(y - f(x)) & , \quad y \geq f(x) \end{cases}$$

şeklinde tanımlanmaktadır. Kantil kayıp fonksiyonu, seçilen γ 'ya göre her örneğin ağırlığını ayarlamaktadır.



Şekil 20: Kantil Kayıp Fonksiyonu

Aşırı tahmin (overestimation) ya da eksik tahmin (underestimation) durumuna göre farklı cezalar uygulamaktadır. Kantil değerinin seçiminde önemli olan, pozitif hataların mı negatif hataların mı daha fazla önemli olduğunu belirlemektir (Wang ve diğerleri, 2020). Bu kayıp fonksiyonunu, sinir ağlarında veya ağaç tabanlı modellerde tahmin aralıklarını hesaplamak için de kullanabilir (Grover, 2018).

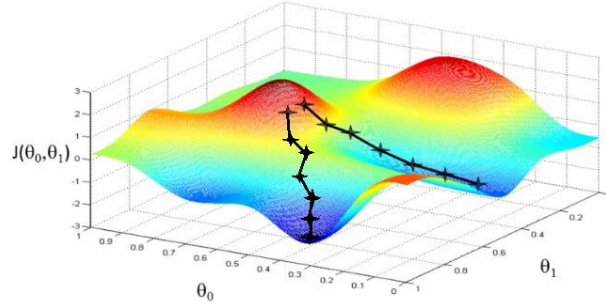
2.6 Optimizasyon Algoritmaları

Derin öğrenmede sıklıkla kullanılan optimizasyon algoritmaları birinci derece metotlar (First Order Methods - FOM) ve ikinci derece metotlar (Second order methods - SOM) olmak üzere iki gruba ayrılmaktadır. Birinci derece metotlar gradyan tabanlı algoritmalarıdır. Birinci derece metotlar içerisinde gradyan iniş algoritması, momentumlu gradyan iniş algoritması, Nestorov momentumlu gradyan iniş algoritması, Adagrad, Adadelta, RMSprop, Adam, AdaMax, Nadam, AMSGrad algoritmaları yer almaktadır. İkinci dereceden metotlar arasında ise Newton, Quasi-Newton (BFGS), Eşlenik gradyan, Hessian-free (Hesse Serbest), Levenberg - Marquardt olmak üzere çeşitli algoritmalar yer almaktadır (Toussaint, 2018). Birinci ve ikinci dereceden metotlarda temelde kullanılan matris yapıları farklıdır. Jacobian matrisi vektörlerin birinci dereceden kısmi türevleri ile elde edilen bir matris ve Hessian matrisi ise fonksiyonun ikinci dereceden kısmi türevlerinin yer aldığı kare matristir. Birinci derece metotlarda Jacobian matrisi kullanılırken, ikinci derece metotlarda Hessian matrisi kullanılmaktadır (Patterson ve Gibson, 2017, s.33). Bu algoritmaların çoğunun temelini gradyan tabanlı algoritmalar oluşturmaktadır. Bu kısımda hem temel algoritma hem de bu algoritmaya bağlı üretilen bazı çözüm algoritmalarına yer verilmiştir.

2.6.1 Gradyan İniş Algoritması

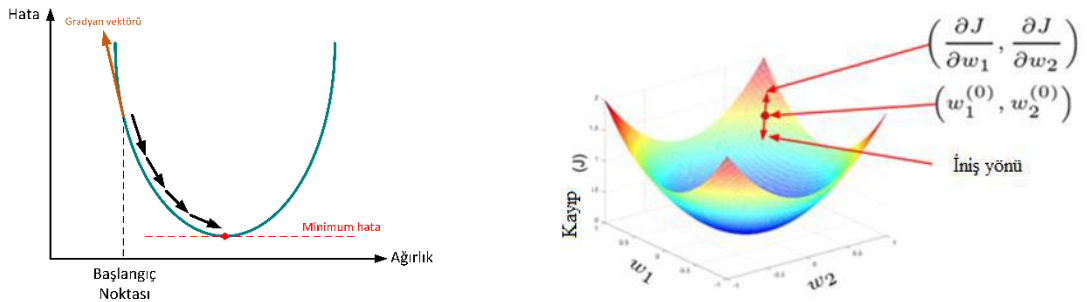
Amaç fonksiyonunu maksimum ya da minimum yapmak makine öğrenmesi algoritmalarının temel hedefidir. Minimum yapılması amaçlanan fonksiyonlar ise kayıp fonksiyonlarıdır. Bu fonksiyon daha önce de bahsedildiği gibi, bir tahmin modelinde gerçek değer ile tahmin edilen değer uygunluğunun bir ölçütüdür. Kayıp fonksiyonun eğiminin negatif olduğu taraftan, en dik inişten iteratif hareketlerle minimum olan

noktaya ulaşmak için kullanılan optimizasyon algoritmasına Gradyan iniş (Gradient Descent) algoritması denir (Ruder, 2016).



Şekil 21: Gradyan İniş Algoritması

Gradyan azalma yönteminde, hata fonksiyonunun “dağlık bir arazi” oluşturduğu düşünüldüğünde, minimum hataya ulaşmak için vadiler içerisinde en alçakta kalan taban noktası bulunmak istenmektedir. Arazideki vadi tabanının enlem ve boylamı, x ve y değerleri (ağırlıklar) olarak ele alınsın. Herhangi bir vadinin tabanı bir yerel minimum noktasını temsil ederken; tüm arazideki en alçakta kalan yer global minimum noktasını temsil etmektedir. Gradyan iniş algoritması ise bu araziye rassal olarak bırakılan bir kişinin mümkün olan en alçak noktaya ulaşması olarak tarif edilebilir. En alçak noktanın elde edilebilmesi için bulunduğu yerin 360 derecesini kontrol eder ve en dik azalma olacak biçimde sonraki adımını atmaktadır. Bu işlem ile bulunan noktalar yerel minimum ya da global minimum olacaktır (Dayhoff ve DeLeo, 2001). İki ağırlıklı sinir hücresinde kayıp fonksiyonunun görseli aşağıdaki şekil gibi olmaktadır.

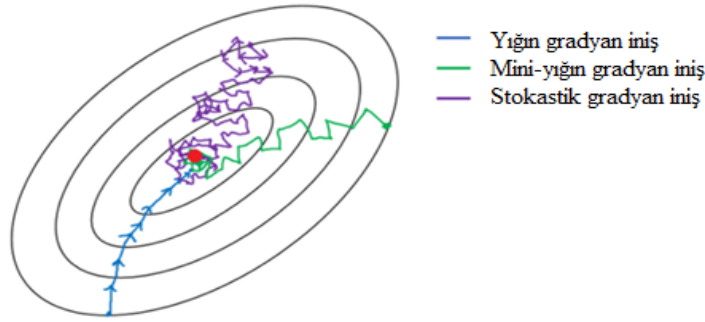


Şekil 22: Gradyan İniş Algoritması Grafiği

Minimize edilmek istenen kayıp fonksiyonu L olmak üzere, eğitim esnasında gerçekleşen her iterasyonda kısmi türevini aldıktan sonra ağırlıklar (w) güncellenmektedir. $\nabla_w L(W)$ kayıp fonksiyonunun gradyanıdır ve

$$\nabla_w L(W) = \left(\frac{\partial L(W)}{\partial W_1}, \frac{\partial L(W)}{\partial W_2}, \dots, \frac{\partial L(W)}{\partial W_d} \right)$$

şeklinde ifade edilmektedir. Daha önce bahsedilen geri yayılım algoritmasında, ağırlıkları güncelleme aşamasında kullanılan yöntemlerden biri gradyan iniş algoritmasıdır. Gradyan iniş algoritmasının üç farklı türü bulunmaktadır. Üç farklı gradyan iniş algoritması için minimum noktaya ulaşırken izlediği yörünge aşağıda gösterilmiştir (Dabbura, 2017).



Şekil 23: Gradyan İniş Algoritmaları için Yörüngeler

Kaynak: Imad Dabbura, “Gradient Descent Algorithm and its Variants”, 2017, <https://towardsdatascience.com/gradient-descent-algorithm-and-its-variants-10f652806a3> (07 Eylül 2021).

Şekil 23’de gösterilen algoritmaların açıklamaları aşağıda yer almaktadır.

2.6.2 Yığın Gradyan İniş

Yığın gradyan iniş (batch gradient descent), vanilya gradyan iniş olarak da adlandırılmaktadır. Veri seti içindeki tüm örnekler kullanılarak ağırlık güncellemeleri sağlanmaktadır. Yığın gradyan inişi,

$$W = W - \alpha \cdot \nabla_w L(W)$$

olarak ifade edilmektedir. Güncelleme işlemi tüm veri seti için gerçekleşeceğinden yığın gradyan inişi yavaş kalabilir. Özellikle büyük bir veri seti ile çalışıldığında algoritmanın

hızının yavaşlayacak olması bu yöntemin dezavantajıdır. Bir diğer dezavantajı ise, modelin çevrimiçi, yani anında yeni örneklerle güncellenmesine olanak sağlamamasıdır (Ruder, 2016). Yığın gradyan inişin avantajları arasında, öğrenme katsayısının sabit olması nedeniyle öğrenme hızını düşürmemesi gelmektedir. Ayrıca minimum noktaya doğru düz bir yörüngeye sahiptir ve kayıp fonksiyonu konveks (dışbükey) ise global minimum noktaya yakınsamasını sağlamakta, aksi takdirde yerel minimuma yakınsamasını sağlamaktadır (Dabbura, 2017).

2.6.3 Stokastik Gradyan İniş

Tüm örneklerin kullanılması yerine, rassal seçilen eğitim örneklerini kullanmaktadır. Stokastik gradyan iniş, ölçülen gradyanların ortalama değerini kullanarak ağ parametrelerini güncellemektedir (Calin, 2020, s.189).

$$W = W - \alpha \cdot \nabla_w L(W, x^{(i)}, y^{(i)})$$

Formülde de görüldüğü gibi, tüm örnekleri gözden geçirmek yerine, her örnekte $x^{(i)}, y^{(i)}$ parametre güncellemesini gerçekleştirir. Bu nedenle, öğrenme her örnekte gerçekleşir. Dikkat edilmesi gereken şey, stokastik gradyan iniş genellikle yığın gradyan inişten daha gürültülü olduğundan, inişindeki rassallık nedeniyle minimuma ulaşmak için genellikle daha fazla sayıda iterasyon gerekmektedir. Ancak bu duruma rağmen, yine de hesaplama açısından yığın gradyan inişe göre daha avantajlıdır. Bu nedenle genellikle, bir öğrenme algoritmasını optimize etmek için yığın gradyan iniş yerine stokastik gradyan iniş tercih edilmektedir (Ruder, 2016).

2.6.4 Mini - Yığın Gradyan İniş

Mini-yığın gradyan iniş algoritması, n tane mini parti oluşturularak eğitim örneği için bir güncelleme gerçekleştirir. Bu işlem gerçekleştirilirken bir önceki sıralamadan etkilenmemek için veri tesadüfi olarak karıştırılmaktadır.

$$W = W - \alpha \cdot \nabla_w L(W, x^{(i:i+n)}, y^{(i:i+n)})$$

Bu yöntem, parametre güncellemelerinin varyansını azaltır, bu da daha kararlı yakınsamaya yol açar. Derin öğrenme kütüphanelerinde ortak olan yüksek düzeyde

optimize edilmiş matris optimizasyonlarından yararlanabilir. Mini-yığın gradyan inişi tipik olarak bir sinir ağını eğitirken tercih edilen algoritmadır ve stokastik gradyan iniş terimi genellikle mini yığınlar kullanıldığında da kullanılır (Dabbura, 2017). Parti boyutunun çok küçük tutulması, hesaplama açısından etkinlik sağlamayacak kadar maliyetli olacaktır. Mini partinin boyutu olarak genellikle donanımsal anlamda en verimli sonucu vermesi için 2'nin katları kullanılır. Yaygın olarak kullanılan değerler 32, 64, 128 veya 256'dır (Aggarwal, 2018, s.123). Yöntemin dezavantajı olarak her döngüde ileri ve geri sıçramalar yapacağından minimum noktaya ulaşamayacak olması söylenebilir. Bu nedenle minimuma yaklaştıkça öğrenme katsayısının düşürülmesi gerekmektedir (Dabbura, 2017).

2.6.5 Uyarlanabilir Moment Tahmini (ADAM)

Derin öğrenmede sık kullanılan optimizasyon algoritmalarından biri Uyarlanabilir Moment Tahmini (ADAM) algoritmasıdır. Gradyan temelli bir algoritma olan ADAM algoritması, daha önce önerilen iki farklı algoritmanın avantajlarını bir araya getirmek üzere tasarlanmıştır. Bunlar Momentum ve RMSprop algoritmalarıdır. Momentum algoritması, Stokastik Gradyan İniş algoritmasında ortaya çıkan salınımları azaltarak hedefe ulaşma hızını arttırmayı amaçlamaktadır. Bunun için, doğrudan gradyanı eklemek yerine, bir önceki adımda elde edilen gradyanın bir oranı eklenerek mevcut ağırlık güncellenir.

$$V_t = \beta V_{t-1} + (1 - \beta) \nabla_w L(W_t)$$

şeklinde tanımlı olmak üzere ağırlıklar:

$$W_{t+1} = W_t - \alpha V_t$$

olarak güncellenir. Burada gradyanlar yerine momentumlu gradyanlar kullanılmaktadır ve V_t 'nin başlangıç değeri 0 ve β ise genellikle 0.9 olarak alınmaktadır.

Momentum algoritması Stokastik Gradyan İnişte hızı arttırmaya yönelik bir yaklaşım iken, RMSprop algoritması ise aynı algoritmadaki sabit öğrenme oranını düzenlemeye yönelik bir yaklaşımdır. RMSprop algoritmasında öğrenme oranı, karesel

gradyanların üstel olarak azalan ortalamasına bölünür ve bu oransal öğrenme oranı kullanılarak mevcut ağırlıklar güncellenir. Bunun için,

$$S_t = \beta S_{t-1} + (1 - \beta)(\nabla_w L(W_t))^2$$

şeklinde tanımlı olmak üzere ağırlıklar,

$$W_{t+1} = W_t - \frac{\alpha}{\sqrt{S_t + \epsilon}} \nabla_w L(W_t)$$

olarak güncellenir. Burada momentumlu gradyanların kareleri kullanılmaktadır ve S_t 'nin başlangıç değeri 0, $\beta = 0.9$ ve $\epsilon = 10^{-6}$ olarak alınmaktadır.

Kingma ve Ba (2015) tarafından önerilen ADAM algoritması, Momentumda elde edilen hedefe hızlı ulaşma ve RMSprop'daki öğrenme oranı avantajlarını birleştirmek üzere tasarlanmıştır. Bunun için Momentum'da yer alan

$$V_t = \beta_1 V_{t-1} + (1 - \beta_1) \nabla_w L(W_t)$$

ve RMSprop'da yer alan

$$S_t = \beta_2 S_{t-1} + (1 - \beta_2)(\nabla_w L(W_t))^2$$

kullanılır. Ancak bu momentlerin özellikle başlangıçtaki adımlarda sıfıra doğru sapmalı olduğu tespit edilmiş ve sapması düzeltilmiş momentlerin kullanımı önerilmiştir:

$$\tilde{V}_t = \frac{V_t}{1 - \beta_1^t}; \quad \tilde{S}_t = \frac{S_t}{1 - \beta_2^t}$$

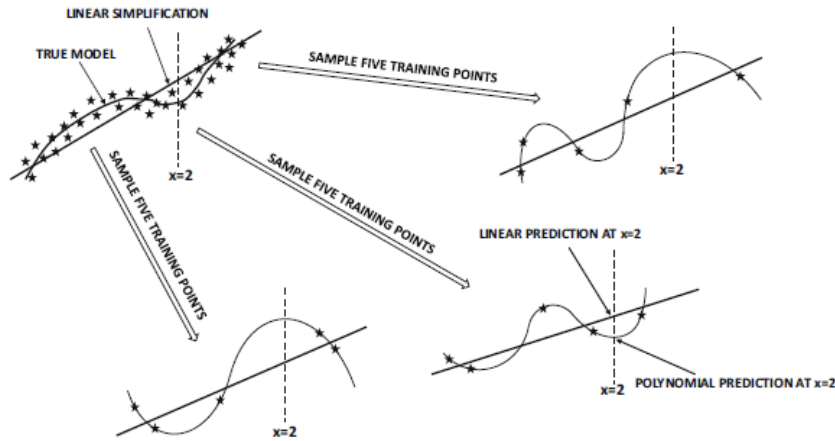
Bu oranlar kullanılarak mevcut ağırlıklar:

$$W_{t+1} = W_t - \frac{\alpha}{\sqrt{\tilde{S}_t + \epsilon}} \tilde{V}_t$$

şeklinde güncellenmektedir. Burada V_t ve S_t 'nin başlangıç değerleri 0, $\alpha = 0.01$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ ve $\epsilon = 10^{-8}$ olarak alınmaktadır (Kingma ve Ba, 2015; Seyyarer ve diğerleri, 2020).

2.7 Eksik - Aşırı Öğrenme, Varyans - Sapma Dengesi, Regülerizasyon

Yapay zeka algoritmalarında veriden öğrenme durumu söz konusudur. Ancak veriden öğrenme işinin hataları beraberinde getirmesi kaçınılmazdır. Makine öğrenmesi algoritmaları kullanılarak tahmin yapılırken çok önemli bir dengenin göz önünde bulundurulması gerekir. Eğitim verisi ile elde edilen modelin veri seti için en uygun model olması istenirken, aynı zamanda verideki tüm aykırı değerleri yakalayan ya da tüm düzensizlikleri açıklayan bir model olması tercih edilmemektedir. Bu durumun *aşırı öğrenme* ve *eksik (yetersiz) öğrenme* olmak üzere iki tür sonucu olabilir. Eğer elde edilen model verilerle çok yüksek uyum içerisindeyse aşırı öğrenme, yüksek bir uyum içerisinde değilse de eksik öğrenme olarak adlandırılır. Modelde değişken sayısının çok olması parametre sayısının fazlaşmasına sebep olacak ve modelin karmaşıklığı artacaktır. Parametre sayısı arttıkça daha doğru tahminler elde edileceği için sapma miktarı azalacak, ancak veriden kaynaklı hatalar olabileceği için varyans artacaktır. Bu noktada dengelenmesi gereken, artan varyans ve azalan sapmadır. Eksik öğrenme daha çok basit modellerde karşılaşılan bir sorunken, aşırı öğrenme karmaşık modellerde karşılaşılan bir sorundur. Özellikle derin öğrenme gibi karmaşık modellerde aşırı öğrenme sıklıkla karşılaşılan bir durumdur (Gürsakal, 2018, s.78).

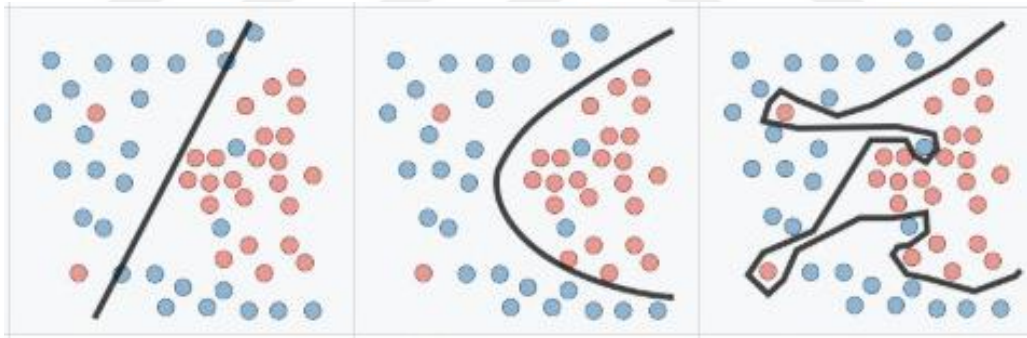


Şekil 24: Artan Model Karmaşıklığı Aşırı Öğrenme

Şekilden anlaşılacağı gibi, yüksek varyansa sahip modeller, eğitim verilerinin rastgele yapılarını öğrenme eğilimindedir, bu durum test örneklerinin tahmininde tutarsızlığa ve hataya neden olur. Daha yüksek dereceli bir polinom modelinin lineer bir modelden daha güçlü olması beklenir. Bu uygulandığında gerekirse çok yüksek dereceli

terimlerin katsayıları 0 olarak alınarak derece düşürülebilir. Ancak bu işlem için yeterince çok örnek olması gerekmektedir. Basitçe söylemek gerekirse, veri kümesinin sonluluğundaki varyans, artan karmaşıklığın ters etki yapmasına neden olur. Modelin gücü ile sınırlı veriler üzerindeki performansı arasındaki uyum varyans-sapma dengesi (Bias-Variance trade-off) ile yakalanır (Aggarwal, 2018, s.171).

Makine öğrenmesi algoritması kullanılırken önce eğitim kümesi örnekleri alınır, sonra bu örnekler ile eğitim hatasını en aza indirgeyecek şekilde parametreler seçilir. Daha sonra da test kümesi örnekleri alınır. Bu süreç boyunca beklenen test hatası ile beklenen eğitim hatasından daha büyük veya eşit olur. Eğitim hatasını küçültme ve eğitim ile test hataları arasındaki farkı azaltma işlemleri algoritmanın ne kadar iyi çalışacağını belirleyen faktörlerdir. Bu iki faktöre göre eksik öğrenme ve aşırı öğrenme sorunları tespit edilir. Model eğitim kümesinde düşük hata oranı üretmediğinde eksik öğrenme, eğitim ve test hataları arasındaki fark çok büyük olduğunda da aşırı öğrenme sorunları ile karşılaşmaktadır (Goodfellow ve diğerleri, 2016, s.109).



Şekil 25: Eksik Öğrenme ve Aşırı Öğrenme

Kaynak: Sebastian Raschka ve Vahid Mirjalili, “Machine Learning and Deep Learning with Python, scikit-learn and Tensor Flow”, Birmingham: Packt Publishing, 2017, s.73.

Şekildeki veri kümesini ikiye ayırmak için doğrusal model yetersiz kalmakta ve yapılan tahminler hata oranını yükseltmektedir. Bu durum ağın eksik (yetersiz) öğrenme gerçekleştirdiğini göstermektedir. Varyansın yüksek olduğu son durumda ise eğitim verisinde tüm detayların ele alınarak çok fazla uyum yakalanmaya çalışıldığı görülmektedir. Ancak bu modelde de tahminler yapıldığında hata oranı yine artış gösterecektir. Bu durum da ağın aşırı öğrenme gerçekleştirdiğini göstermektedir (Raschka ve Mirjalili, 2017, s.73).

Aşırı öğrenmeden kaçınmak için, veri çoğaltma, erken durdurma, girdi karmanı ve gizli katmandaki bazı düğümleri analizden çıkarma (dropout, bırakma) ve regularizasyon yapılabilir (Gürsaka, 2017, s.198).

Bu yöntemler kullanılırken verilerin eğitim, doğrulama ve test etme için kullanılması gerekir. Model oluşturulurken verilerin tümü kullanılamaz. Çünkü hem model oluşturma hem de test etme için aynı veri setini kullanmak aşırı uyum oluşturmaktır. Ayrıca, model seçimi ve parametre ayarlama için kullanılan veri kümesi bölümünün de model oluşturma için kullanılanlardan farklı olması gerekir. Yaygın bir hata da hem parametre ayarlama hem de son değerlendirme (test) için aynı veri setini kullanmaktır. Böyle bir yaklaşım, eğitim ve test verilerini kısmen karıştırır ve hataya sebep olur. Belirli bir veri seti her zaman verilerin kullanım şekline göre tanımlanan üç bölüme ayrılmalıdır (Aggarwal, 2018, s.178):

Eğitim Verisi: Verilerin bu kısmı sinir ağının ağırlıklarını öğrenme sürecini oluşturmak için kullanılır. Modelin oluşturulması sırasında öğrenme oranı veya düzenleme için farklı hiperparametreler ile farklı model tasarımları mevcut olabilir. Ayrıca, modeller farklı algoritmalar üzerinden birden çok kez denenebilir. Eğitim verileri ile, farklı modellerden en iyi algoritmanın seçildiği model belirlenir. Ancak en iyi modeli seçmek için bu algoritmaların gerçek değerlendirmesi, eğitim verileri dışında ayrı bir doğrulama veri seti üzerinde yapılır.

Doğrulama Verisi: Verilerin bu kısmı, model seçimi ve parametre ayarı için kullanılır. Örneğin, öğrenme oranı seçimi, modelin veri setinin (yani eğitim verilerinin) ilk kısmı üzerinde birden çok kez oluşturulması ve ardından bu farklı modellerin doğruluğunu tahmin etmek için doğrulama setinin kullanılmasıyla ayarlanabilir. Doğrulama verileri, algoritmanın parametrelerini ayarlamak veya en iyi tasarım seçimini seçmek için bir tür test verisi seti olarak görülmelidir.

Test Verisi: Verilerin bu kısmı, nihai (ayarlanmış) modelin doğruluğunu test etmek için kullanılır. Aşırı öğrenmeyi önlemek için parametre ayarlama ve model seçimi sürecinde test verilerine bakılmaması bile önemlidir. Test verileri, sürecin en sonunda

yalnızca bir kez kullanılır. Test veri setinin yalnızca bir kez kullanılması oldukça katı bir gerekliliktir.

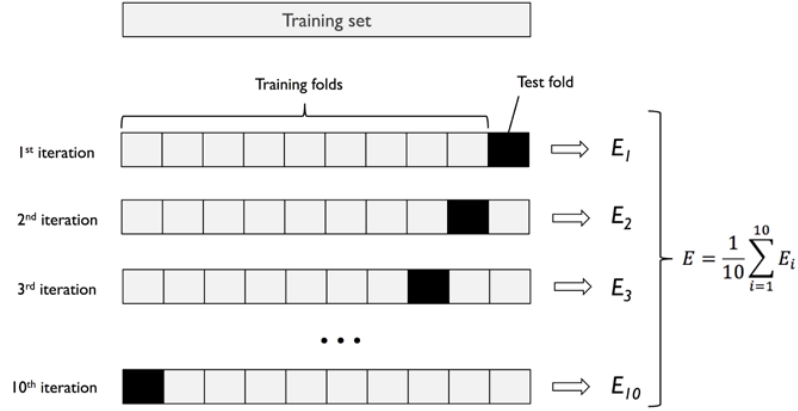
Veri kümesi eğitim için %70, test için %30 olacak şekilde bölünmesi önerilmektedir. Eğitim algoritması esnasında, eğitim verisinin bir kısmı eğitim için, bir kısmı doğrulama için kullanılmaktadır. Veri kümesinin büyüklüğüne bağlı olarak %70 eğitim, %30 doğrulama ya da %80 eğitim, %20 doğrulama için kullanılabilir (Heaton, 2015, s. 257).

2.7.1 Çapraz Geçerlilik (Doğrulama)

Basit anlamda çapraz geçerlilik (cross validation) yöntemini ilk olarak Kurtz 1948 yılında önermiştir. 1951 yılında Moiser tarafından geliştirilmiş ve çift çapraz geçerliliği ortaya koymuştur. 1982 yılında da Krus ve Fuller tarafından çoklu çapraz geçerlilik teknikleri geliştirilmiştir (Kurtz, 1948; Mosier, 1951; Krus ve Fuller, 1982).

Çapraz geçerlilik yönteminde modelin doğruluğu test edilmektedir. Bazı öğrenme algoritmaları için kullanıcının belirleyeceği hiperparametrelere ihtiyaç duyulmaktadır. Genellikle çeşitli hiperparametre değerleri kullanılarak doğruluğu en yüksek modelin belirlenmesinde çapraz geçerlilik kullanılmaktadır (Gürsakal, 2018, s.70). Veri seti sınırlı olduğunda, eğitim, doğrulama ve test kümelerine bölmek için yeterli miktarda veri olmadığından çok küçük kümeler oluşmaktadır. Küçük doğrulama ve test kümeleri ise ağırlık düşük performans göstermesine neden olacaktır. Bu duruma çözüm getirmek için K-katlı çapraz doğrulama (K-fold Cross-Validation) tekniği kullanılmaktadır (Bishop, 2006, s.32).

Çift çapraz geçerlilik ve çoklu çapraz geçerlilik, veri setinin sınırlı sayıda olması durumunda kullanılmaktadır. Bu yöntemler, modelleme aşamasında bütün veri setinin kullanımına imkân sağlamaktadır. Veri seti, bir bölümü eğitim verisi diğer bölümü test verisi olmak üzere tesadüfî olarak iki eşit parçaya ayrılır ve bu şekilde bir doğruluk hesaplaması yapılır. Ardından test verisi ile eğitim verisi yer değiştirilerek yeniden bir doğruluk hesaplaması yapılır ve doğruluk değerlerinin ortalaması alınarak modelin doğruluk oranı hesaplanır. Bu algoritma çift çapraz geçerlilik olarak tanımlanmaktadır (Temel ve diğerleri, 2012).



Şekil 26: K-katlı Çapraz Geçerlilik

K-katlı çapraz geçerlilik yöntemi ise, her verinin bir kez eğitim bir kez de test verisi olarak kullanabileceği geçerlilik yöntemlerinin genel halidir. Bu yöntemde veri seti, k eşit gruba ayrılmakta ve bir grup test olarak, geriye kalan (k-1) grup ise eğitim verisi olarak kullanılmaktadır ve bu durum k kez tekrar edilir. Yine doğruluk değerlerinin ortalaması alınarak modelin doğruluk oranı belirlenir. Modellerin ortalama performansı bu bağımsız katlara bağlı olarak hesaplandığından verilerinin alt bölümlere ayrılmasına daha az duyarlıdır (Raschka ve Mirjalili, 2017, s.191; Hastie ve diğerleri, 2008, s.241). Her parçalanmada bir parça test verisi olarak ayrılırken diğer parça eğitim verisi olarak alınır ve gerçek y değeri ile tahmin edilen $f(x)$ değeri karşılaştırılarak modellerin performansı değerlendirilir. Ayrıca modellerin her birinden tahmini test hatası hesaplanır ve model seçimi yapılır (Gürsakal, 2018, s.71).

2.7.2 Regülerizasyon Teknikleri

Aşırı öğrenme sorununa çözüm getirmek amacıyla yaygın kullanılan regülerizasyon teknikleri şunlardır:

Veri Çoğaltma: Aşırı öğrenmeden kaçınmak için kullanılan yöntemlerden biri veri çoğaltmaktır. Ancak yeniden veri toplamak her zaman mümkün olmayacağı için genellikle tercih edilen bir yöntem değildir. Bunun yerine mevcut verilerle yapılan veri çoğaltma (Data Augmentation) teknikleri kullanılmaktadır. Görüntü işlemede verileri çoğaltmak için mevcut resimlerin simetrisi alınmakta, ters çevrilmekte, rengi, kontrastı,

ölçeği değiştirilmekte, gürültü eklenmekte, bulanıklaştırılma, resmi kırpma, bazı kısımları silme gibi teknikler uygulanmaktadır (Ghandi, 2021).

Erken Sonlandırma: Eğitim verisi ile test verisi arasındaki fark açıklığı belli bir seviyeye geldiğinde eğitimi durdurur. Bazı algoritmalar bunu sürekli kontrol ederek otomatik olarak yapar. Genellikle ağın ilk eğitimlerinde hem eğitim hem test hataları azalacak, ancak bir noktada ağ eğitim verilerini “ezberlemeye” başlayacaktır. Bu durumda eğitim verisi için hata azalırken, test verisi tam tersine artmaya başlayacaktır. Bu noktada aşırı öğrenmeyi engellemenin bir yolu uygun noktada eğitimi durdurmaktır (Albon, 2018, s.316).

L1 Regülerizasyonu: LASSO (en küçük mutlak küçültme ve seçme operatörü) olarak da adlandırılmaktadır. İlk olarak Tibshirani (1996) tarafından ortaya atılan bir değişken seçme ya da değişken azaltma yöntemi olarak kullanılır. LASSO Regresyon yönteminin Lagrange formundaki fonksiyonu;

$$L = \sum_{i=1}^n \left(y_i - \sum_{j=0}^m x_{ij} w_j \right)^2 + \lambda \sum_{j=0}^m |w_j|$$

olarak gösterilir. Burada n gözlem sayısı, m ağırlık sayısı, λ negatif olmayan ayarlama parametresidir. Burada ağırlıkların mutlak değeri cezalandırma işleminde kullanılır. L1 cezalandırma normu olarak kullanılan bu yöntem, bazı ağırlık tahminlerini λ yeterince büyük olduğunda sıfır olmaya zorlar ve elemine eder. Böylece aşırı öğrenme engellenmiş olur.

L2 Regülerizasyonu: Ridge regresyonu olarak da bilinmektedir. Hata kareler toplamını minimize eden katsayıları, bu katsayıları bir ceza uygulayarak bulmaktır. Aşırı öğrenme sorununa karşı kullanılmaktadır. Çok boyutluluğa çözüm sunan bir yöntemdir. Tüm değişkenler ile model kurar, ilgisiz değişkenleri çıkarmaz sadece katsayılarını sıfıra yaklaştırır. Modeli kurarken λ (ceza) için iyi bir değer bulmak gerekir.

$$L = \sum_{i=1}^n \left(y_i - \sum_{j=0}^m x_{ij} w_j \right)^2 + \lambda \sum_{j=0}^m w_j^2$$

biçiminde ifade edilir (Hastie ve diğerleri, 2008; Gürsakal, 2018).

Elastik Net Regülerizasyonu: L1 ve L2 regülasyon tekniklerinin kombinasyonu ile çalışmaktadır.

$$L = \sum_{i=1}^n \left(y_i - \sum_{j=0}^m x_{ij} w_j \right)^2 + \lambda_1 \sum_{j=0}^m |w_j| + \lambda_2 \sum_{j=0}^m w_j^2$$

şeklinde ifade edilir.

Seyreltme (Dropout): Ağın aşırı uyumunu önlemek için eğitim esnasında gizli katmandaki belirli sayıda nöronun rassal olarak bırakılmasıdır. Tüm girdinin tek seferde gönderilmesi yerine girdiler rassal seçilmiş eşit büyüklükte birbirinden bağımsız yığılara bölünerek ağda eğitilirler (Yılmaz ve Kaya, 2019). Bırakma, sinir ağlarını düzenlemek için popüler ve güçlü bir yöntemdir. Bırakmada, eğitim için bir grup gözlem oluşturulduğunda, bir veya daha fazla katmandaki birimlerin bir oranı sıfırla çarpılır (yani bırakılır). Her grup aynı ağ üzerinde eğitilir, ancak her grup, o ağın mimarisinin biraz farklı bir versiyonuyla karşı karşıya kalır. Bırakma işlemi, her partide birimleri sürekli ve rastgele bırakarak, birimleri farklı ağ mimarileri altında parametre değerlerini öğrenmeye zorlaması bakımından etkili bir işlemdir. Böylece gizli birimlerdeki gürültülere karşı sağlam olmayı öğrenirler ve bu, ağın eğitim verilerini basitçe ezberlemesini engelleyerek aşırı öğrenmenin önüne geçer. Bırakma, hem gizli hem de girdi katmanlarına mümkündür (Albon, 2018, s.318).

3. DERİN ÖĞRENME ALGORİTMALARI

Derin öğrenme algoritmaları farklı özelliklere sahip katmanların ve bu katmanların içerisinde yer alan alternatif yapıların eklenmesiyle çeşitlilik kazanmaktadır. Bu bölümde temel derin öğrenme algoritmalarından Konvolüsyonel Sinir Ağları ve Yinelemeli Sinir Ağları algoritmalarının detaylarına yer verilecektir.

3.1 Konvolüsyonel Sinir Ağları

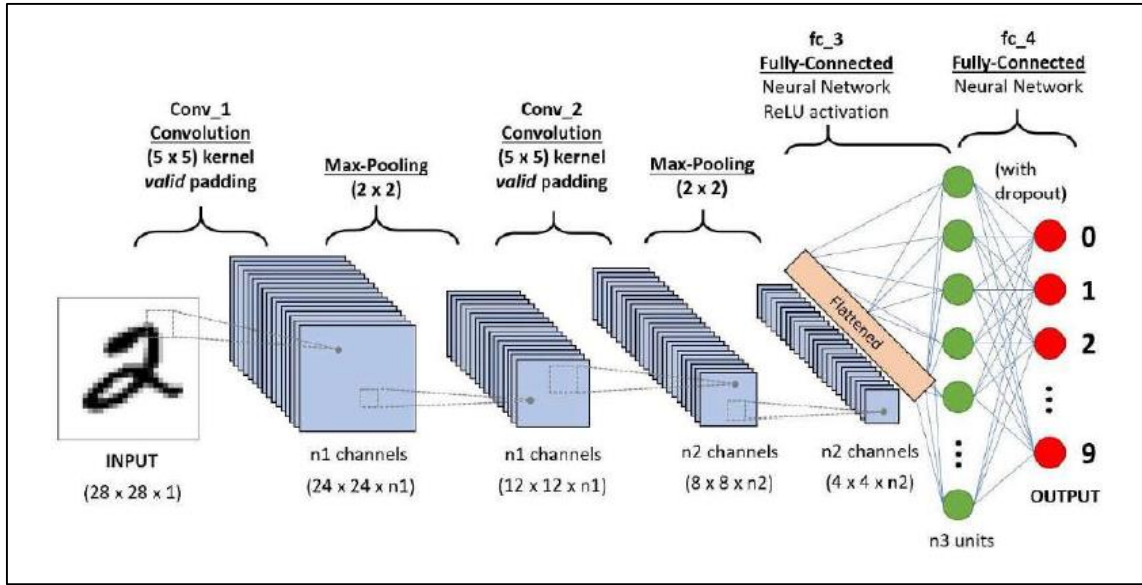
Konvolüsyonel (evrişimsel) sinir ağları (Convolutional neural network, CNN) modern derin öğrenme algoritmaları içerisinde en temel ağ mimarilerinden biridir. CNN, nesne tanıma, görüntü işleme, doğal dil işleme, sınıflandırma gibi amaçlarla yaygın olarak kullanılan bir sinir ağıdır. Bu ağ girdi olarak alınan verileri işleyerek gerekli özellikleri/öznitelikleri bu verilerden otomatik olarak çıkarabilen bir yapıya sahiptir.

CNN mimarisi, temelleri sinirbilim çalışmalarına dayanan, uzun ve detaylı bir geçmişe sahiptir. Hubel ve Wiesel tarafından sinirbilim alanında yapılan çalışmalar bu yapıların temellerini oluşturan öncü çalışmalar olarak kabul edilmektedir (Hubel ve Wiesel, 1959). Hubel ve Wiesel memeli görüşünün nasıl çalıştığını anlamak amacıyla kedilerin sinir hücreleri üzerinde detaylı incelemeler yapmışlardır. Görü sisteminde gelen görüntünün beyinde işlenmeye başlandığı ilk yer V1 adıyla bilinen birincil görsel kortektir. Bilgisayar görüşü ile memeli görüşü arasında bu alanda yapılan işlemler açısından benzerlikler bulunduğu tespit edilmiştir. Bu benzerliklerden ilki görüntünün iki boyutlu olarak işlenmesidir. Bir diğer benzerlik ise V1 alanında bulunan basit hücreler tarafından gerçekleştirilen belli bir alandaki görüntünün doğrusal bir fonksiyonla temsil edilmesi özelliğidir. Benzer durum CNN içerisinde bulunan katmanlarda da gerçekleştirilmektedir. V1’de çok sayıda bulunan karmaşık hücreler tarafından gerçekleştirilen işlem ise CNN’de bulunan havuzlama (pooling) katmanıya benzerlik göstermektedir. Bu benzerlikler dikkate alındığında, memelilerdeki görü sistemi CNN algoritmasında baz alınan fikrin alt yapısını oluşturmuştur (Goodfellow ve diğerleri, 2016, s.366).

CNN algoritması günümüzde kullanılan modern yapısına ulaşana kadar çeşitli aşamalardan geçmiştir. İlk zamanlarda kullanılan modellerde geri yayılım ve gradyan iniş gibi algoritmalar yer almamaktadır. Öncü görü modellerinden biri olan “neocognitron”, geleneksel anlamda konvolüsyon katmanı ve öznitelik çıkarma gibi fikirleri içeren denetimsiz bir algoritmadır (Fukushima, 1980). Sonrasında Lang ve Hinton (1988) zaman gecikmeli sinir ağlarında (TDNN) geri yayılım algoritmasından faydalanmışlardır. Zaman serileri için kullanılan geri yayılım algoritmasının başarılı sonuçlar vermesi ile birlikte, bu algoritma LeCun ve diğerleri (1989) tarafından görüntü verisi için kullanılmaya başlanmıştır. Böylece modern CNN mimarisi ortaya çıkmıştır. Olshausen ve Field (1996) tarafından önerilen seyrek kodlama yapısının basit hücrelere benzer olarak öznitelikleri öğrenebildiği tespit edilmiştir.

3.1.1 CNN Mimarisi

Klasik bir yapay sinir ağında girdi tek boyutlu sütun vektörü olarak ele alınır. Girdi katmanından sonra tam bağlantılı bir ya da birden fazla gizli katman kullanılarak çıktı katmanına aktarım gerçekleşir. Her bir katmanda bulunan nöronlar arasındaki tam bağlantı yapısından kaynaklı olarak parametre sayısında bir artış söz konusu olacaktır. Bu parametre artışı ve ağın derinliğinin artması aşırı öğrenme sorununa sebep olacaktır. Aşırı öğrenme sorunu, ağın eğitimi için kullanılan verilerde başarılı bir öğrenme çıktısı vermekle birlikte eğitim dışında kullanılan verilerde başarısız sonuçlar vermesine sebep olacaktır. Bu sorunlara karşı çözüm oluşturan bir ağ mimarisi olarak konvolüsyonel sinir ağları kullanılmaktadır. Konvolüsyonel sinir ağlarında girdi, klasik bir yapay sinir ağı aşamasından önce farklı katmanlarda işlem görmektedir. Klasik bir CNN mimarisi genel olarak iki aşamadan oluşmaktadır. İlk aşamada görüntülerden elde edilmesi planlanan belirleyici özellikler tespit edilmektedir. İkinci aşamada ise bu özelliklerin sınıflandırma problemi kapsamında ele alınan sınıflara atama işlemi gerçekleştirilir. İlk aşamada konvolüsyon (convolution), ReLU ve havuzlama (pooling) katmanları kullanılarak özellik haritaları belirlenmektedir. İkinci aşamada ise klasik yapay sinir ağı yapısındaki gibi tam bağlantılı (fully-connected) katman, ReLU ve seyreltme (drop-out) katmanları yer almaktadır. Klasik bir CNN mimarisi Şekil 27’de yer almaktadır.



Şekil 27: CNN Mimarisi Genel Yapısı

Şekil 27’de görüldüğü gibi, girdi olarak alınan görüntü önce konvolüsyon katmanında işlem görmektedir. Bu katman, özelliklerin tespitindeki en önemli katman olup, çeşitli boyutlarda ağırlık matrisleri yardımıyla özellik çıkarımı yapılan katmandır. Filtre ya da kernel olarak da adlandırılan bu ağırlık matrislerinin girdi üzerinde adım adım kaydırılmasıyla özellik haritaları elde edilir. Elde edilen özellik haritaları ReLU gibi bir aktivasyon fonksiyonu yardımıyla negatif değerlerden arındırılır. Sonraki katman olan havuzlama katmanında ise çıkarılan özellik haritaları daha düşük boyutlu hale getirilerek hesaplama avantajı sağlamaktadır. Ayrıca havuzlama katmanı, benzer özellikleri bir araya getirme işlevine sahiptir. Özellik haritasının çıkarılmasında kullanılan bu katmanlar birden fazla kez tekrarlanabilir. Bu tekrarlı yapı ağı derinliğini belirlemektedir (LeCun ve diğerleri, 2015).

İkinci aşamada ise matris şeklinde ancak ilk haline göre boyutu azaltılmış olan özellik haritası düzleştirme (flatten) işlemi ile vektör haline getirilir. Başlangıçta görüntü olan girdi bu aşamada vektör haline geldikten sonra, artık klasik yapay sinir ağı işlemleri gerçekleşmektedir. Burada tam bağlantılı katmanlar arası geçiş sağlanır. Aktivasyon fonksiyonu olarak genellikle ReLU tercih edilir. Bu noktada bazı bağlantıların pasif hale getirilmesi yoluyla düğüm seyreltme işlemi yapılır ve böylece ağı ezberlemesi önlenir. Son aşamada da sınıflandırma işleminde başarılı olan softmax aktivasyon fonksiyonu kullanılarak ilgili özelliklerin belirlenen sınıflara atanması işlemi gerçekleştirilir. Burada

softmax dışında farklı aktivasyon fonksiyonlarının kullanılması mümkündür (Yılmaz ve Kaya, 2019, s.66). CNN mimarisinde sözü edilen katmanların detaylarına bu bölümün devamında yer verilecektir.

CNN algoritması klasik yapay sinir ağları yapısından önemli derecede farklılıklar göstermektedir. Bu farklılıklar CNN'nin yapay sinir ağlarına göre avantajlı taraflarının temelini oluşturmaktadır. CNN'nin doğal yapısında bulunan üç özellik diğer ağlardan farklılaşmasında etkin rol oynamaktadır. İlk özellik *seyrek etkileşim* olarak adlandırılmaktadır. Bu özellik katmanlar arasındaki bağlantıların yapısı ile alakalıdır. YSA'da giriş katmanı ile diğer katmanlar arasında tam bağlantılı bir yapı vardır. Bir başka ifade ile her bir girdi birimi kendinde sonra gelen her birime bağlandığı için tamamen bağlı bir yapı söz konusudur. CNN'de ise her bir girdi biriminin her bir çıktı birimine bağlanmasına ihtiyaç duyulmamaktadır. Her bağlantıyı her katmanda kullanmak ciddi bir maliyet ve hesaplama karmaşıklığı yaratacağından algoritmanın çalışması verimsiz hale gelecektir. Bunun yerine önemsiz bağlantılar alınmayarak etkileşimin daha yüksek olduğu bağlantılar dikkate alınmaktadır. Bu seyreltme işlemi ile ortaya çıkma ihtimali olan bilgi kayıpları, daha sonra ara katmanların daha sık kullanılmasıyla dolaylı olarak ağ akışı içerisinde giderilmektedir. Ayrıca karmaşık etkileşimlere sahip yapılar basit yapı bloklarıyla tanımlanarak daha etkin bir şekilde modellenmeye olanak sağlamaktadır.

CNN'nin klasik sinir ağlarına göre avantajlı olmasını sağlayan ikinci özellik ise *parametre paylaşımı* olarak bilinmektedir. Klasik bir YSA'da her bir girdi için farklı ağırlıklandırma parametreleri uygulanırken, CNN'de girdilerin tümü için ortak ağırlıklandırma parametreleri belirlenmektedir. YSA ile CNN arasındaki bu yaklaşım farkı, hesaplanması gereken parametre sayının büyük ölçüde azalmasını sağlamaktadır. Örneğin, 28x28 piksele sahip bir görüntü girdi olarak alındığında, bir girdi için $28 \times 28 = 784$ adet ağırlık parametresi hesaplanır. Ara katmanda 30 nöron olduğu durumda, $30 \times 784 = 23520$ adet ağırlık ve 30 adet sapma parametresi ile birlikte toplam 23550 parametrenin hesaplanmasına ihtiyaç duyulmaktadır. CNN'de ise örneğin 5x5 boyutunda bir ortak ağırlıklandırma matrisi kullanıldığında bir girdi için $5 \times 5 = 25$ adet ağırlık parametresi hesaplanır. Ayrıca ortak olarak da bir sapma parametresi kullanılacağından toplam 26 parametre hesaplanması gerekir. Çıkarılması gereken 20 özellik haritası olduğu

durumda, hesaplanması gereken parametre sayısı $20 \times 26 = 520$ olacaktır. Klasik YSA'daki parametre sayısına göre CNN'de çok daha az parametrenin hesaplanması yeterli olmaktadır. Bu parametre paylaşımı özelliği, CNN algoritmasına istatistiksel verimlilik ve ihtiyaç duyulan bellek kapasitesi anlamında önemli bir avantaj sağlamaktadır (Nielsen, 2015, s.173).

Seyrek etkileşim ve parametre paylaşımı özellikleri diğer ağlara göre CNN'nin nispeten daha kolay eğitim sürecine sahip olmasına olanak sağlamaktadır. Bu özellikler görüntü verisinde sıklıkla kullanılan CNN algoritmasının, zaman serileri için kullanımında da avantaj sağlamaktadır (Kaya ve Ögüdücü, 2020).

CNN algoritmasını diğer ağlardan ayıran üçüncü özellik ise *eşit değişirlik* özelliğidir. Bu özellik, girdi biriminde yapılan küçük değişikliklerin aynı şekilde çıktı birimine de yansımaları sağlamaktadır. Bir başka ifade ile bu özellik sayesinde girdide yapılan değişiklikler çıktı biriminde de korunmaktadır. Eşit değişirlik özelliği matematiksel olarak,

$$f(g(x)) = g(f(x))$$

şeklinde ifade edilmektedir. Burada, g fonksiyonu x girdisi üzerinde değişiklik yapan bir fonksiyon olmak üzere, x girdisini g fonksiyonu ile değiştirip ardından f fonksiyonunu uygulamak ile önce f fonksiyonu uygulayıp daha sonra çıktıyı g fonksiyonu ile değiştirmek aynı sonucu verecektir. Örneğin, görüntü verisi ele alındığında piksel kaydırma işlemini yapıp sonra konvolüsyon adımını uygulamak, konvolüsyon işleminden sonra piksel kaydırmakla eşdeğer olacaktır. Görüntü verisi ele alındığında, ilgilenilen nesnenin yeri değişse dahi eşit değişirlik özelliği gereğince algoritma nesneyi yine de tespit edecektir. Bu özelliğin uygulanabilmesi parametre paylaşımı özelliğinden kaynaklanmaktadır (Goodfellow ve diğerleri, 2016, s.339). Ayrıca eşit değişirlik kavramının özel bir hali de değişmezlik özelliğidir. Değişmezlik özelliği girdide oluşacak herhangi bir değişime karşı çıktının değişmemesidir. Bu özellik girdideki önemsiz değişimlere karşı sonucun değişmemesini mümkün kılmaktadır. Değişmezlik özelliği CNN algoritmasında havuzlama katmanında ortaya çıkmaktadır. Havuzlama işlemi, görüntü dönüşümlerine yönelik uzamsal değişmezlik özelliğini arttırmaktadır (Singh ve

diğerleri, 2020). Ancak geleneksel CNN mimarisi dönüşümlere karşı eşit değişirli olmasına rağmen girdideki ölçek değişimlerine karşı eşit değişirli değildir. Örneğin, girdi olarak alınan bir görüntüde yer alan nesnenin boyutu değiştiğinde algoritmadan elde edilen çıktıda aynı değişim yakalanamamaktadır. Bu soruna alternatif bir çözüm olarak Sosnovik ve diğerleri (2020) tarafından yönlendirilebilir filtreler ile ölçek-eşit değişirli bir yapı önerilmiştir. Böylece girdideki ölçek değişimlerine karşı sonuçların korunduğu bir algoritma ortaya konmuştur.

3.1.2 Konvolüsyon Katmanı

Bir CNN mimarisinde temel katman konvolüsyon katmanıdır. Bir sinir ağının Konvolüsyonel sinir ağı olarak ele alınabilmesi için en az bir konvolüsyon katmanı içermesi beklenmektedir. Bu katmanda özellik çıkarımı için filtreler kullanılmaktadır. Bu filtreler boyutu değişebilen ağırlık matrisleridir. Filtrelerin boyutu 2x2, 3x3, 5x5, 7x7 vb. şekilde seçilebilir. Boyutu belirlenen filtre, matris şekline getirilen görüntü verisi üzerinde adım adım gezdirilir. Filtrenin görüntü verisi üzerinde eşleştiği her bir yerel alana *alıcı (reseptif) alan* denmektedir. Adım büyüklüğü kullanılan verinin ve filtrenin boyutuna göre değişiklik gösterebilir. Filtre görüntü üzerinde gezdirilirken her bir adımda matris skaler çarpımı kullanılarak bir doğrusal birleşim işlemi yapılır. Bu işleme *konvolüsyon (evrişim)* işlemi adı verilir.

Görüntü verisine konvolüsyon işleminin uygulanması fikrinin temeli, memeli görüsünün matematiksel alt yapısına dayanmaktadır. Burada kullanılan yapı, biyolojik sinir hücresine gelen görüntünün konumuna göre ağırlıklandırılması işlemine benzemektedir. Memelilerin görsel korteksindeki basit hücrelerin, görüntüyü alıcı alanına göre ağırlıklandırmak için kullandığı fonksiyonun, Gabor (1946) tarafından tanımlanan fonksiyona oldukça benzediği tespit edilmiştir (Jones ve Palmer, 1987). Gabor fonksiyonu,

$$w(x, y) = \alpha \exp\left(-\frac{\hat{x}^2 + \gamma \hat{y}^2}{2\sigma^2}\right) \cos\left(2\pi \frac{\hat{x}}{\lambda} + \psi\right)$$

$$\hat{x} = (x - x_0) \cos(\theta) + (y - y_0) \sin(\theta)$$

$$\hat{y} = -(x - x_0) \sin(\theta) + (y - y_0) \cos(\theta)$$

olarak tanımlanmaktadır. α ölçek parametresini, λ kosinüsün dalga boyu faktörünü, θ Gabor fonksiyonunun yönünü, ψ fazın ofset değerini ve γ ise uzamsal görüş açısını belirtmektedir. Ayrıca x_0 , y_0 ve θ bir koordinat sistemi tanımlamaktadır (Imamura ve Arizumi, 2021).

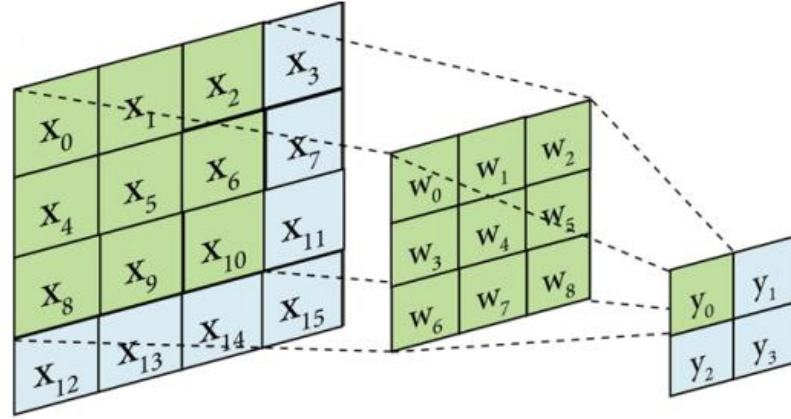
Buna göre V1 basit hücrenin, $I(x, y)$ konumundaki görüntüye verdiği tepki

$$s(I) = \sum_x \sum_y w(x, y) I(x, y)$$

şeklinde ifade edilebilir. Burada $w(x, y)$ Gabor ağırlıklandırma fonksiyonudur. Memeli görüntüsünde görüntüyü konumuna göre ağırlıklandırmak için kullanılan bu fonksiyonun benzeri CNN algoritmasında konvolüsyon aşamasında gerçekleşmektedir. İki boyutlu I görüntüsü için konvolüsyon işlemi,

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(m, n) K(i - m, j - n)$$

olarak ifade edilmektedir. Burada K , kernel ağırlıklandırma fonksiyonudur ve memeli görüşündeki $w(x, y)$ Gabor fonksiyonunun işlevine karşılık gelmektedir (Goodfellow ve diğerleri, 2016, s.333). Burada kernel ağırlıklandırma işlemi filtreler yardımıyla yapılmaktadır. Dolayısıyla konvolüsyonda filtre ile kernel kavramları birbirleri yerine kullanılmaktadır. Burada yapılan işlem bir skaler çarpımdır, dolayısıyla doğrusal birleşime denk gelmektedir. Konvolüsyon işlemi Şekil 28’de gösterilmektedir.



Şekil 28: Konvolüsyon İşlemi

Kaynak: Lukas Mosser, Olivier Dubruleve ve Martin J. Blunt., “Stochastic Reconstruction of an Oolitic Limestone by Generative Adversarial Networks”, *Transp Porous Media*, 125, 2017, s.81-103.

Şekil 28’de, x girdi matrisini, w ağırlıklandırma matrisi filtreleri, y matrisi ise özellik haritasını göstermektedir. Kullanılan filtrenin 3x3 boyutunda olması sebebiyle, x girdi matrisindeki 3x3 boyutundaki taralı alan alıcı alana karşılık gelmektedir. Özellik haritasının ilk elemanı olan y_0 ,

$$y_0 = x_0w_0 + x_1w_1 + x_2w_2 + x_4w_3 + x_5w_4 + x_6w_5 + x_8w_6 + x_9w_7 + x_{10}w_8$$

skaler çarpımıyla hesaplanmaktadır. Bu w filtresi adım adım gezdirilerek her adımda benzer şekilde hesaplama yapılır ve özellik haritası çıkarılır. Filtrenin adım adım gezdirilmesi sol üstten başlanarak yapılmaktadır. Ancak bu durumda kenarda yer alan pikseller filtrelemenin merkezine denk gelmeyeceğinden konvolüsyon işlemi uygulanamamaktadır. Bu durum kenar piksellerinde var olması muhtemel olan önemli bilgilerin kaybolmasına neden olmaktadır. Bu sorunun çözümü için dolgu (padding) işlemi gerçekleştirilmektedir. Örnek bir dolgu işlemi Şekil 29’da verilmektedir.

0	0	0	0	0	0
0	x_0	x_1	x_2	x_3	0
0	x_4	x_5	x_6	x_7	0
0	x_8	x_9	x_{10}	x_{11}	0
0	x_{12}	x_{13}	x_{14}	x_{15}	0
0	0	0	0	0	0

Şekil 29: Dolgu İşlemi

Şekil 29’da görüldüğü gibi dolgu işlemi kenar piksellerinin etrafına 0 satır ve/veya sütunlarının eklenmesiyle gerçekleştirilmektedir. Böylece kenar piksellerinin filtrenin merkezine denk gelmesi mümkün olmaktadır. Böylece bilgi kaybı önlenmiş olur (Chollet, 2018, s.125). Bu işleme özel olarak sıfır dolgu (zero padding) adı verilmektedir. Farklı türde dolgu çeşitleri de mevcuttur.

Konvolüsyon katmanında belirlenmesi gereken üç önemli hiperparametre bulunmaktadır. Bu parametreler; filtrenin boyutu, adım büyüklüğü ve dolgunun yapılıp yapılmayacağıdır. Bu hiperparametreler yardımıyla özellik haritası çıktısının boyutu,

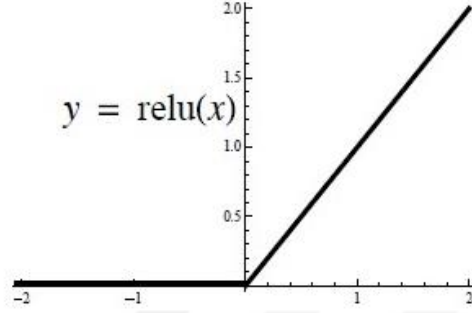
$$\text{Çıktının Boyutu} = \frac{\text{Girdinin boyutu} - \text{Filtrenin boyutu} + 2 * (\text{Dolgu sayısı})}{\text{Adım Sayısı}} + 1$$

olarak hesaplanır (Bezdan ve Baćanın Džakula, 2019).

CNN mimarisinde konvolüsyon katmanından sonra gelen havuzlama katmanına geçilmeden önce ara bir aktivasyon işlemi yapılmaktadır. Bu aşamada, özellik haritaları çıkarıldıktan sonra elde edilen özniteliklerin bir sonraki aşamada aktif olarak kullanılıp kullanılmayacağı belirlenmektedir. CNN algoritmalarında sıklıkla kullanılan aktivasyon fonksiyonları ReLU, sigmoid ve tanh fonksiyonlarıdır. Bunlardan sigmoid ve tanh doğrusal olmayan aktivasyon fonksiyonları arasında olup, ağ yapısı ilerledikçe gradyanlarının sıfır olması gibi sorunlar nedeniyle genellikle ReLU aktivasyon fonksiyonu tercih edilmektedir. ReLU aktivasyon fonksiyonu,

$$f(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases}$$

olarak tanımlanmaktadır. ReLU aktivasyon fonksiyonunun grafiği Şekil 30'da verilmiştir.



Şekil 30: ReLU Aktivasyon Fonksiyonu

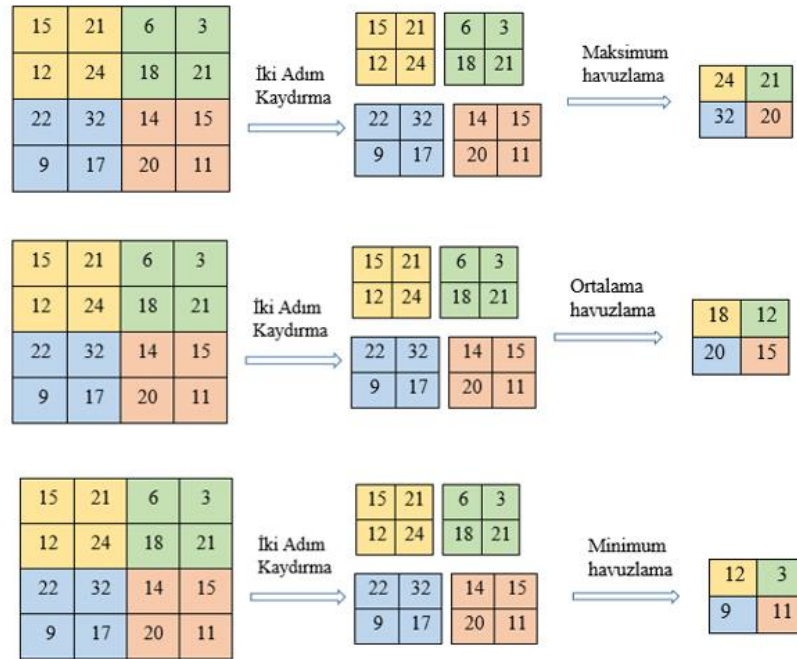
Fonksiyonundan da anlaşıldığı gibi girdideki negatif değerleri sıfıra çevirirken, pozitif değerleri aynı şekilde korumaktadır. Böylece ağ yapısında her özelliğin bir sonraki katmana aktarımı gerçekleşmeyecektir. Dolayısıyla ağdaki önemli olan özelliklerin aktarımı yapılacaktır. Bu durum da derin sinir ağlarının eğitiminde hesaplama kolaylığı sağlayarak, öğrenme hızını yükseltecektir (Patterson ve Gibson, 2017, s.131).

3.1.3 Havuzlama Katmanı

Konvolüsyon katmanından sonra havuzlama (pooling) katmanı gelmektedir. Havuzlama katmanından önce aktivasyon işlemi yapılarak özellik haritası çıkarılır. Bu özellik haritasını alt örneklemesini yapmak için havuzlama katmanı kullanılmaktadır (Shen ve diğerleri, 2017). Havuzlama katmanı filtrenin taradığı bölgedeki görsel tepkinin özetini sunmaktadır (Goodfellow ve diğerleri, 2016, s.343). Çıkarılan özellik haritalarının havuzlama katmanında boyutu azaltılarak bir sonraki katmana geçiş sağlanır. Başka bir ifade ile havuzlama katmanında kendinden sonra gelecek olan konvolüsyon katmanına daha az satır ve sütun elemanı içeren çıktılar üretmek amaçlanmaktadır. Böylece havuzlama katmanında filtreler özellik haritaları üzerinde seçilen adım büyüklüğüne göre gezdirilerek özellik matrisinin uzamsal boyutu azaltılmaktadır. Bir CNN modelinde birden fazla kez havuzlama katmanı kullanılarak aşamalı olarak veri indirgenmektedir. Bu boyut azaltma işlemi tüm ağın eğitiminde kullanılacak parametre sayısında azalışa sebep olacak, dolayısıyla işlem hızı artacaktır. Ayrıca girdinin boyutunda gerçekleşen bu

azaltma işlemi aşırı öğrenme problemine karşı da çözüm olmaktadır. CNN’de ardışık olarak uygulanan konvolüsyon katmanları arasına havuzlama katmanının yerleştirilmesi sıklıkla karşılaşılan bir uygulamadır (Patterson ve Gibson, 2017, s.266; İbrahim ve diğerleri, 2019, s.124).

Derin öğrenme modellerinde farklı havuzlama yöntemleri uygulanmakla birlikte genel olarak havuzlama katmanı üç farklı şekilde yapılır. Bu yöntemler; maksimum havuzlama (max pooling), ortalama havuzlama (average pooling) ve minimum havuzlama (min pooling) yöntemleridir. Bu üç yöntem arasından maksimum havuzlama uygulamada daha sık kullanılmaktadır. *Maksimum havuzlama* yönteminde, belirlenen filtre boyutuna göre girdiyi çakışmayan bölgelere ayırarak, her alt bölgede yer alan en yüksek değeri çıkış birimi olarak ele almaktadır. *Ortalama havuzlama* yönteminde, birbiriyle çakışmayacak ayrılmış bölgelerde bulunan her alt bölgede bulunan elemanların ortalaması alınmaktadır. *Minimum Havuzlama* yönteminde ise benzer olarak alt bölgelerde bulunan matris elemanlarından minimum olanı temsili olarak alınmaktadır (Shyam, 2021). Havuzlama işlemine ait farklı yöntemlere ilişkin örnek gösterim Şekil 31’de yer almaktadır.



Şekil 31: Maksimum, Ortalama ve Minimum Havuzlama Örneği

Şekil 31’de 4x4 boyutlu bir girdi matrisine 2x2 boyutlu filtre 2 adım büyüklüğü ile gezdirilmiştir. Havuzlama işlemi gerçekleştirildikten sonra çıktı olarak 2x2 boyutlu bir matris elde edilmiştir. Havuzlama katmanında işlem görmüş bir görüntünün çıktı boyutu, G_1 girdinin genişliği, Y_1 girdinin yüksekliği, D_1 girdinin derinliği, F filtre boyutu ve A adım büyüklüğü olmak üzere,

$$\text{Çıktı Boyutu} = G_2 \times Y_2 \times D_2$$

$$G_2 = \frac{(G_1 - F)}{A} + 1$$

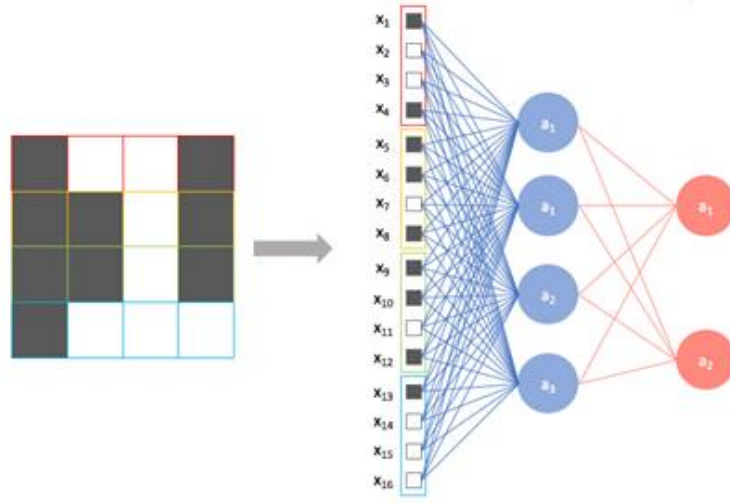
$$Y_2 = \frac{(Y_1 - F)}{A} + 1$$

$$D_2 = D_1$$

olarak ifade edilmektedir. Havuzlama işleminde uygulamada genel olarak $F = 2$ ve $A = 2$ alınarak işlem yapılmaktadır (İnik ve Ülker, 2017).

3.1.4 Düzleştirme ve Tam Bağlantılı Katman

Konvolüsyon ve havuzlama katmanları tamamlandıktan sonra tam bağlantılı (fully connected) katman uygulanmaktadır. Bu katman yoğun (dense) katman olarak da bilinmektedir. Ancak havuzlama katmanında özellik matrisi elde edildikten sonra tam bağlantılı katmana geçilmeden önce *düzleştirme (flatten)* işlemi yapılmaktadır. Bu işlem, görüntü girdisine uygulanan bir dizi konvolüsyon ve havuzlama işlemlerinden sonra elde edilen matris şeklindeki yapı tek boyutlu bir vektöre (ya da sayı dizisi) dönüştürme işlemi olarak tanımlanmaktadır. Bu işlemle birlikte bir önceki katmanda çıkarılan tüm özellikler tek boyutlu bir yapı şeklini almaktadır. Örneğin 5x5x3 şeklinde üç boyutlu özellik matrisi 75 elemanlı tek boyutlu bir vektöre dönüşmektedir (Shyam, 2021). Düzleştirme ve tam bağlantılı katman için örnek bir görsel Şekil 32’de yer almaktadır.



Şekil 32: Düzleştirme İşlemi ve Tam Bağlantılı Katman

Kaynak: Jeremy Jordan, “Convolutional Neural Networks”, 2017, <https://www.jeremyjordan.me/convolutional-neural-networks/> (21 Ağustos 2022).

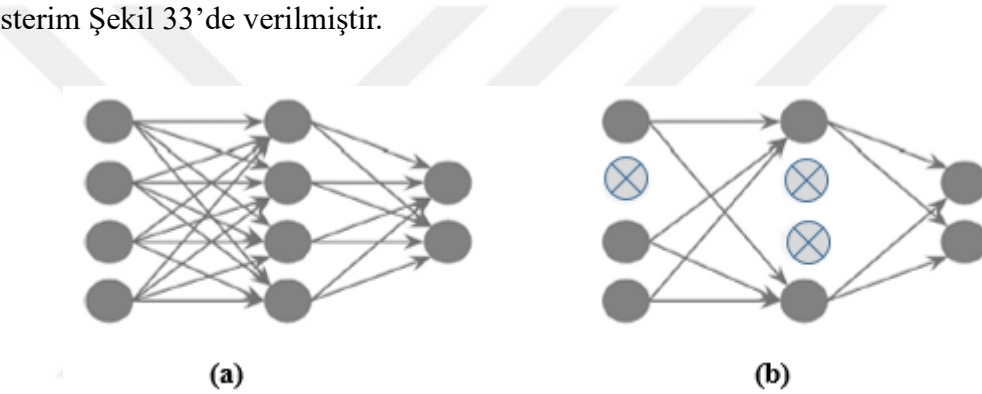
Düzleştirme işlemi gerçekleştikten sonra klasik sinir ağı yapısı adımları işlemeye başlamaktadır. Burada her girdinin her çıktıya bağlandığı bir veya daha fazla katman kullanıldığı tam bağlantılı yapı söz konusudur. Tam bağlantılı katmanın üzerinde yer alan her bir nöron belirli ağırlıklarla çarpılarak her bir çıkış nöronuna bağlanmaktadır. Tam bağlantılı katmanda, çıkarılan özelliklerin araştırmanın çıktısı olarak bilinen hangi sınıfla en iyi şekilde ilişkili olduğunu tespit edilmektedir. Bu katmanın son aşamasında sınıf sayısı ile eşleşecek şekilde çıktı oluşmaktadır (Yamashita ve diğerleri, 2018).

3.1.5 Toplu Normalleştirme ve Seyreltme

Konvolüsyonel sinir ağları oluşturulurken konvolüsyon katmanı, havuzlama katmanı, tam bağlantılı katman dışında, olası bazı sorunların önüne geçebilmek adına, modele toplu normalleştirme (batch normalization) ve düğüm seyreltme (dropout) katmanları da eklenebilmektedir. Aşırı öğrenme, derin öğrenme modellerinde sıklıkla karşılaşılan sorunlardan biridir. Aşırı öğrenme, modelin eğitimi esnasında yüksek öğrenme oranlarına ulaşılmasına rağmen test verisi ile işlem yapıldığında aynı başarı yakalanamaması olarak ifade edilmektedir. Toplu normalleştirme ve düğüm seyreltme işlemleri, bu sorunun önlemek amacıyla kullanılan işlemlerdir. Ayrıca bu iki işlem, aşırı öğrenmenin yanı sıra modelin yakınsama süresini azaltarak daha hızlı çalışmasını sağlamaktadır (Garbin ve diğerleri, 2020).

Toplu normalleştirme işlemi, klasik normalizasyondaki gibi sadece girdi katmanında yapılan işlemin aksine, her katman arasında bu işlemin yapılmasına imkan sağlamaktadır. Toplu normalleştirme stokastik gradyan algoritmasında mini gruplarla çalışmakta ve aktivasyonları normalleştirmek sadece o grubun ortalamasını ve varyansını kullanmaktadır. Bu işlem sayesinde ağıdaki katmanlar, önceki katmanın öğrenmesini beklemek zorunda kalmadan hızlı bir şekilde eğitimi gerçekleştirmektedir (Ioffe ve Szegedy, 2015; Bjorck ve diğerleri, 2018).

Seyreltme işlemi, rassal olarak seçilen bazı nöron/düğüm setlerinin eğitim aşaması sırasında yok sayılması anlamına gelmektedir. Seyreltme işlemine ait örnek bir gösterim Şekil 33’de verilmiştir.



Şekil 33: Standart Sinir Ağı (a) ve Seyreltme İşlemi Yapılan Sinir Ağı (b)

Kaynak: Christian Garbin, Xingquan Zhu ve Oge Marques, “Dropout vs. Batch Normalization: An Empirical Study of Their Impact to Deep Learning”, *Multimedia Tools and Applications*, 79(19), 2020, s.12777-12815.

Seyreltme işleminde, önem derecesi az olan birimlerin modele dahil edilmemesiyle, modelin hem öğrenme hızının hem de öğrenme başarısının artması sağlanmaktadır. Konvolüsyonel sinir ağlarında seyreltme işleminin tam bağlantılı katmanda uygulanması başarılı sonuçlar vermektedir (Hinton ve diğerleri, 2012; Krizhevsky ve diğerleri, 2012).

3.1.6 Sınıflandırma Katmanı

Tam bağlantılı katmandan sonra çıktıyı oluşturan son aşama sınıflandırma katmanıdır. CNN modelinde tam bağlantılı katmanın çıktısındaki vektör yapılarının uygun sınıflara atanması gerekmektedir. Vektörlerdeki birimlerin uygun sınıflara atanması için uygun bir aktivasyon fonksiyonu kullanılmalıdır. Sınıflar arası farkı daha

açık bir şekilde ortaya koyması sebebiyle genellikle softmax fonksiyonu kullanılmaktadır. Ayrıca çoklu sınıflandırma problemlerinde de kullanım kolaylığı sağlaması sebebiyle tercih edilmektedir. Softmax fonksiyonu girdi vektörünün çıktı olarak belirlenen sınıflara ait olma olasılıklarını hesaplar. Softmax fonksiyonu,

$$f(x_i) = \frac{e^{x_i}}{\sum_{j=1}^N e^{x_j}}$$

$$F(x_i) = \max_{i \in [1, N]} f(x_i)$$

olarak tanımlanmaktadır. Burada x_i tam bağlantılı katmandan elde edilen değerleri göstermek üzere, $f(x_i)$ her bir i değeri için sınıflara ait olma olasılığını göstermektedir. Birimler, en büyük olasılık değerine sahip sınıfa atanmaktadır (Wu ve diğerleri, 2016).

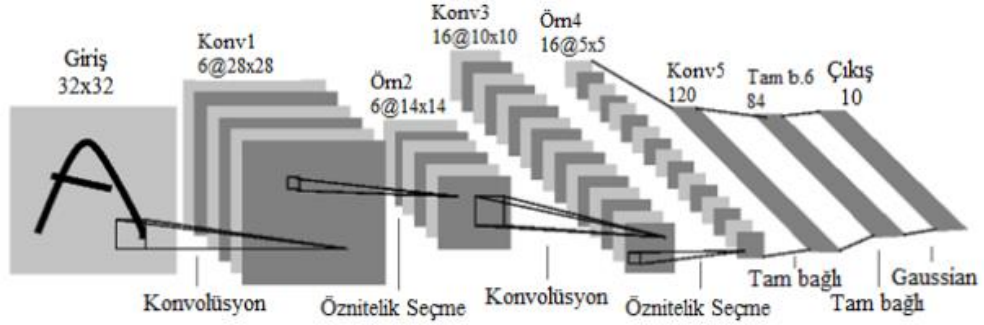
3.2 Diğer Konvolüsyon Sinir Ağı Mimarileri

Temel bir konvolüsyon sinir ağı mimarisinde birkaç konvolüsyon katmanı, bunu takip eden havuzlama katmanı, ardından bir ya da birkaç tane evrişim katmanı ve başka bir havuzlama katmanı olmak üzere derinleşen bir yapı bulunmaktadır. Görüntü girdisi ağ ilerledikçe küçülmekle birlikte, daha fazla özellik haritası oluşturarak kapsamı artar. Bu adımı tam bağlantılı katman takip ederek ileri beslemeli klasik sinir ağına geçilir ve son katman ile sınıflandırma gerçekleşir. Son yıllarda temel CNN mimarisinde ilerlemeler kaydedildi ve alternatif mimariler geliştirildi. Bu ilerlemenin temelini, ImageNet Büyük Ölçekli Görsel Tanıma Yarışması (The ImageNet Large Scale Visual Recognition Challenge, ILSVRC) kapsamında elde edilen başarılı mimariler oluşturmaktadır (Geron, 2017, s.365). Klasik mimarilerden LeNet 5, AlexNet, VGGNet; modern mimarilerden GoogleNet ve ResNet en popüler CNN mimarileri arasında yer almaktadır (Bezdan ve Baćanin Džakula, 2019).

3.2.1 LeNet-5

1989 yılında LeCun ve arkadaşları tarafından geliştirilen LeNet mimarisi başarılı olarak uygulanan ilk ağ yapısıdır. El yazması rakamlardan oluşan MNIST veri setinde sınıflandırma uygulaması olarak ele alınmıştır. Daha sonra Yann LeCun ve ekibi

tarafından 1998’de LeNet modeli geliştirilerek hata oranı daha az olan Şekil 34’de yer alan yapı oluşturulmuştur.



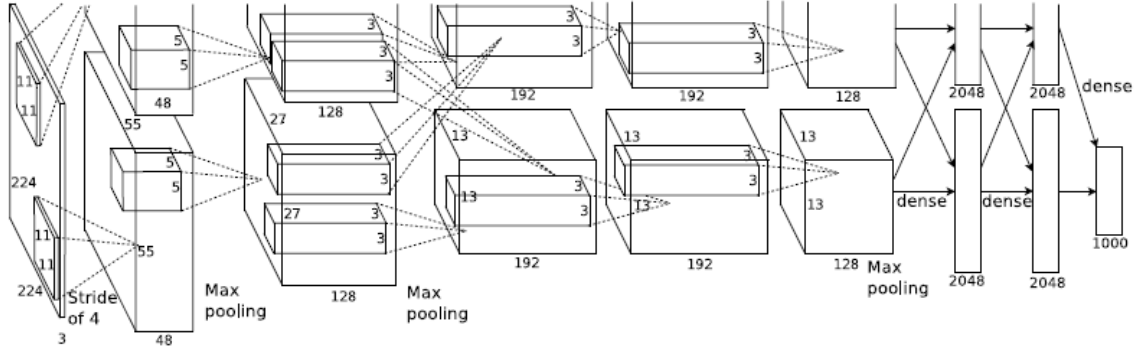
Şekil 34: LeNet-5 Mimarisi

Kaynak: Yann LeCun, Leon Bottou, Yoshua Bengio ve Patrick Haffner, “Gradient-Based Learning Applied to Document Recognition”, *Proceedings of the IEEE*, 86(11), 1998, s. 2278-2324.

Şekil 34’de gösterildiği gibi, mimari iki evrişim katmanına, iki ortalama havuzlama katmanına, iki tam bağlantılı katmana ve Gauss bağlantılı bir çıktı katmanına sahiptir. LeNet-5 mimari toplam 60.000 parametreye sahiptir. Ara katmanlarda aktivasyon fonksiyonu olarak hiperbolik tanjant aktivasyon fonksiyonu, son katmanda ise softmax aktivasyon fonksiyonu kullanılmıştır (LeCun ve diğerleri, 1998).

3.2.2 AlexNet

2012 yılında Alex Krizhevsky ve ekibi tarafından geliştirilen AlexNet modeli, %15,4 hata oranı ile ILSVRC yarışmasında birinci olarak dikkat çeken bir ağ mimarisi oldu. AlexNet, LeNet-5 mimarisinden geliştirilmesine rağmen, ardışık konvolüsyon katmanları kullanımıyla ve daha derinleşen yapısıyla LeNet-5’ten ayrılmaktadır. Ayrıca AlexNet’te, LeNet-5’te hiperbolik tanjant aktivasyon fonksiyonu yerine ReLU fonksiyonu kullanılmıştır. 1,2 milyon yüksek çözünürlüklü görüntüyü 1000 farklı sınıfa sınıflandırmak için derin bir ağ yapısı oluşturulmuştur. AlexNet mimari yapısı Şekil 35’de yer almaktadır.



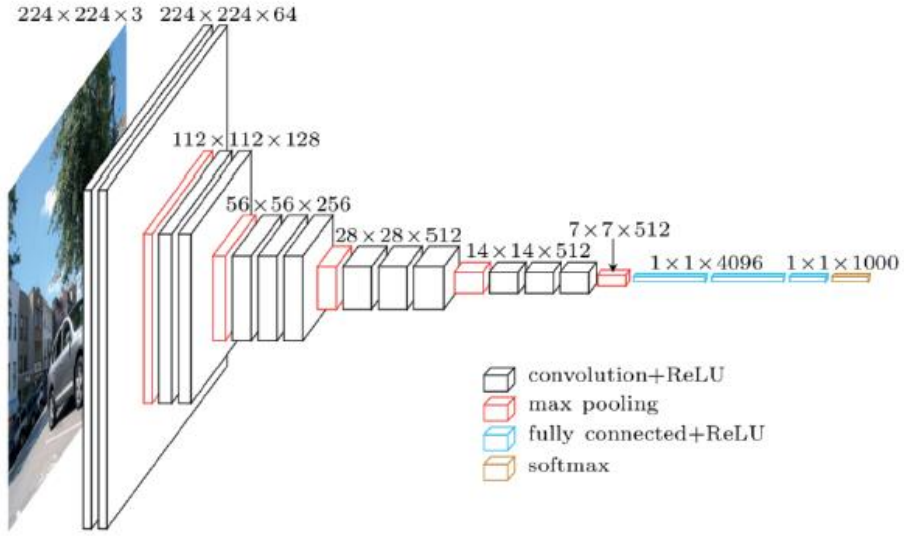
Şekil 35: AlexNet Mimarisi

Kaynak: Alex Krizhevsky, Ilya Sutskever ve Geoffrey E. Hinton, “Imagenet Classification With Deep Convolutional Neural Networks”, *Advances in Neural Information Processing Systems*, F. Pereira, C. J. Burges, L. Bottou ve K. Q. Weinberger, (Ed.). 25 (NIPS 2012), 2012.

Şekil 35 incelendiğinde, AlexNet mimarisinin 5 konvolüsyon katmanı, 3 maksimum havuzlama katmanı ve 2 tam bağlantılı katman katmandan oluştuğu görülmektedir. Eğitimi daha hızlı hale getirmek ve aşırı öğrenme soruna çözüm oluşturmak adına tam bağlantılı katmanda ise seyreltme işlemi uygulanmıştır. AlexNet’te toplam kullanılan parametre sayısı 60 milyondur (Krizhevsky ve diğerleri, 2012).

3.2.3 VGGNet

2014 yılında Karen Simonyan ve Andrew Zisserman tarafından geliştirilen VGGNet mimarisi, diğer mimarilere göre daha derinleşen bir ağ yapısına sahiptir. Bu ağ yapısı ile, ağın derinliğinin daha başarılı sınıflandırma doğruluğu elde etmek için kritik bir bileşen olduğunu tespit edilmiştir. AlexNet’e göre doğruluk oranı daha yüksek olsa da 138 milyon parametre içermesi sebebiyle fazla miktarda bellek kullanımına ihtiyaç duyulmaktadır. Şekil 36’da VGGNet mimarisi yer almaktadır.



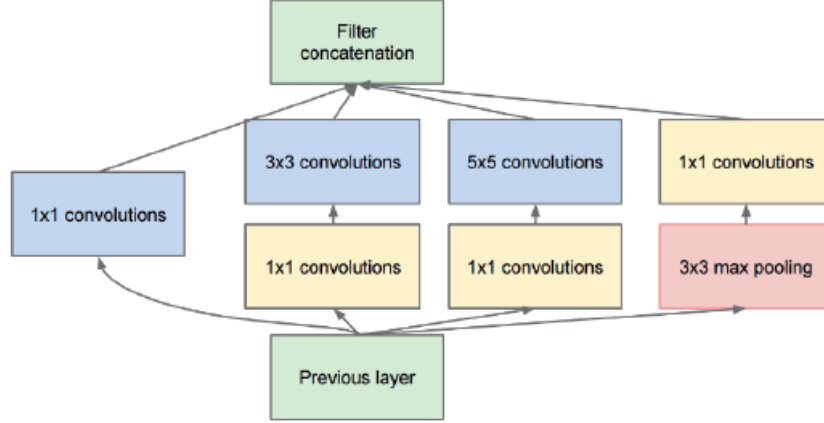
Şekil 36: VGGNet Mimarisi

Kaynak: Timea Bezdan, ve Nebojsa Baćanin Džakula, “Convolutional Neural Network Layers and Architectures”, *Paper presented at Sinteza 2019 - International Scientific Conference on Information Technology and Data Related Research*, 2019.

Şekil 36’da görüldüğü gibi, VGGNet mimarisinde 13 konvolüsyon, 5 maksimum havuzlama ve 3 tam bağlantılı katman kullanılmaktadır (Simonyan ve Zisserman, 2014).

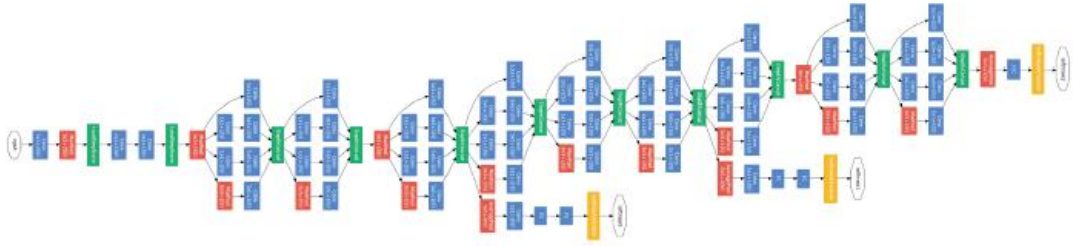
3.2.4 GoogleNet

GoogleNet mimarisi 2014 yılında Christian Szegedy ve arkadaşları tarafından geliştirilmiştir. Bu mimari yapıda, yalnızca konvolüsyon ve havuzlama katman olmayıp, “başlangıç (inception)” adı verilen modüller bulunmaktadır. Bu modül Şekil 37’de verilmiştir.



Şekil 37: Başlangıç (inception) Modülü

Bu modüller, CNN mimarisinde ilk defa kullanılmasıyla modern CNN mimarisi için öncü niteliği taşımaktadır. GoogleNet'i karmaşık bir yapı haline getirmektedir. %6,7 hata oranına sahip olan bu mimari VGGNet mimarisine göre daha hızlı çalışmaktadır. Başlangıç modülünde, konvolüsyon katmanlarındaki 1×1 , 3×3 , 5×5 filtrelerin ve 3×3 maksimum havuzlama işleminin eş zamanlı olarak gerçekleştirilmektedir. Bu yapıya naive başlangıç modülü adı verilmektedir. GoogleNet mimarisi Şekil 38'de gösterilmektedir.



Şekil 38: GoogleNet Mimarisi

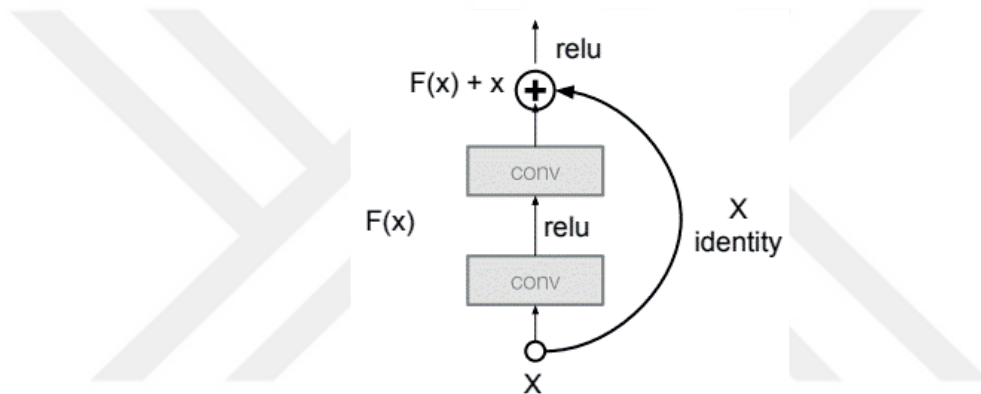
Kaynak: Cristian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke ve Andrew Rabinovich, "Going deeper with convolutions", In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, s. 1-9

Şekil 38'de görüleceği gibi 9 adet başlangıç modülü bulunmaktadır. GoogleNet'te tam bağlantılı katmanlar kullanılmamıştır. Böylece kullanılan parametre sayısını oldukça azaltmıştır. Kullanılan parametre sayısına göre AlexNet ile

kıyaslandığında yaklaşık 10 katı daha az parametre kullanımı söz konusudur (Szegedy ve diğerleri, 2015).

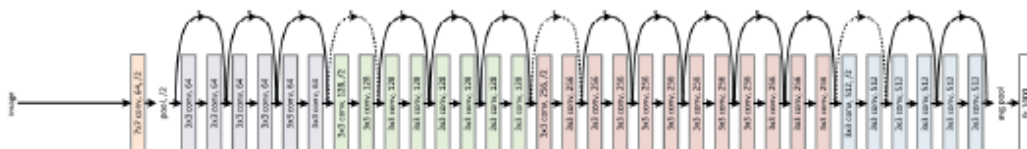
3.2.5 ResNet

2015 yılında Kaiming He ve ekibi tarafından geliştirilen ResNet mimarisi, Artık Sinir Ağı olarak da adlandırılmaktadır. ResNet mimarisinin temelini artık bloklar oluşturmaktadır. Artık bloklar, eğitim sırasındaki kazanılan bilgilerin korunmasına ve ağ kapasitesinin artırılmasına imkan sağlamaktadır. Bu bloklar, ardışık katmanlar birbirine bağlanırken kullanılmaktadır. Artık blok yapısı Şekil 39’da verilmiştir.



Şekil 39: Artık Blok Yapısı

Şekil 39’da görüldüğü gibi, artık bloklarla X girdisinin bir veya birden fazla katmanını atlayarak ileri taşınması işlemini gerçekleştirilmektedir. Artık bloklarda, x girişi konvolüsyon-ReLU, konvolüsyon serisinden sonra bir $F(x)$ sonucu üretmektedir. Bu sonuç daha sonra orijinal x girişine eklenerek $H(x) = F(x) + x$ olarak ifade edilir. 34 Katmanlık örnek bir ResNet mimarisi Şekil 40’da yer almaktadır.



Şekil 40: ResNet Mimarisi

Kaynak: Kaiming He, Xiangyu Zhang, Shaoqing Ren ve Jian Sun, “Deep Residual Learning for Image Recognition”, In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, s. 770-778.

ILSVRC yarışmasında ResNet mimarisi, diğer ağlara göre daha derin bir yapıya sahip olup, 152 katmandan oluşmaktadır. Bu mimari %3,6 ile oldukça düşük bir hata oranına sahip olması sebebiyle büyük bir başarı göstermiştir (He ve diğerleri, 2015).

3.3 Yinelemeli Sinir Ağları

Yapay zekanın gelişiminde insan beyninin çalışma yapısı temel alınmaktadır. İnsan beyninin temel özelliklerinden bir tanesi geçmiş bilgileri hatırlaması ve gelecekteki çıkarımlar için bu bilgileri işlemesidir. Yinelemeli sinir ağları (Recurrent Neural Network, RNN) geçmiş bilgileri hatırlaması ve sıralı veri bilgisini dikkate alması fikrine dayanmaktadır. RNN’de sıra yapısının dikkate alınmasıyla hafıza özelliği ortaya çıkmaktadır. RNN’nin bu özelliği, doğal dil işleme (NLP) ve duygu analizi başta olmak üzere, zaman serileri, finansal veriler, video ve ses tanıma vb. alanlarda yaygın olarak tercih edilen bir derin öğrenme mimarisi olmasını sağlamıştır (Geron, 2017, s.379).

Yinelemeli sinir ağlarının temelini geri yayılım algoritması oluşturmaktadır. Ancak geri yayılım algoritmasından önce Hopfield (1982) tarafından RNN’nin özel bir türü olan Hopfield ağları önerilmiştir. Sonrasında Rumelhart ve diğerleri (1986) tarafından ortaya konan geri yayılım algoritması yinelemeli sinir ağlarının temelini oluşturmuştur. Daha sonra RNN’nin özel türleri geliştirilmiştir. Bunlar arasında Jordan ağları (Jordan, 1986), Elman ağları (Elman, 1990), zaman gecikmeli sinir ağları (Lang ve diğerleri, 1990) ve yankı durum ağları (Jaeger, 2001) gibi örnekler bulunmaktadır. Ancak Hochreiter ve Schmidhuber (1997) tarafından geliştirilen Uzun-Kısa Süreli Bellek Ağları (LSTM) ve Cho ve diğerleri (2014) tarafından geliştirilen Geçitli Yinelenebilir Birimler (GRU) günümüzde en sık kullanılan RNN türleridir.

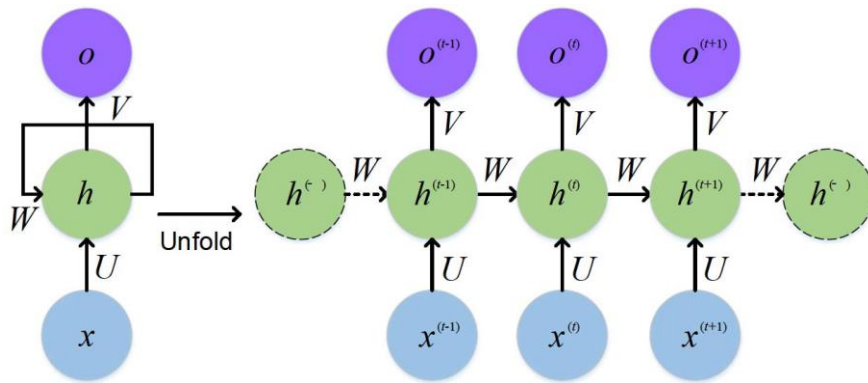
Yinelemeli sinir ağları geçmişi dikkate almasıyla diğer sinir ağlarından farklılaşmaktadır. Klasik bir ileri beslemeli sinir ağında girdi birimlerinden çıktı birimlerine birebir eşleme yapılırken, yinelemeli sinir ağlarında girdi birimlerinin geçmişinden her çıktıya eşleşme yapılmaktadır. Burada önemli olan nokta önceki girdilerin bilgisini içeren bir belleğin oluşması ve bu bellek sayesinde geçmiş bilgilerin ağ yapısı boyunca çıktıyı etkilemesine izin verilmesidir (Graves, 2012, s.22). Konvolüsyonel sinir ağları, görüntü verisindeki gibi yükseklik ve genişlik boyutunun ele

alındığı uzamsal yapılar için avantaj sağlamaktadır. Ancak konvolüsyon işleminde sıra yapısı dikkate alınmamaktadır (Goodfellow ve diğerleri, 2016, s.374).

CNN mimarisini klasik sinir ağlarından ayıran temel özelliklerden bir tanesi parametre paylaşımı özelliğidir. RNN algoritmasında CNN'ye benzer olarak parametre paylaşımı özelliği bulunmaktadır. Ancak iki mimari arasında parametre paylaşımı bakımından farklı yaklaşımlar söz konusudur. CNN'de girdi birimlerinin komşuluğunda bir ağırlıklandırma yapılır ve tüm ağ boyunca aynı ağırlıklandırma parametreleri kullanılmaktadır. Başka bir ifade ile konvolüsyon işleminin çıktı birimleri girdilerin bir fonksiyonudur. RNN'de ise çıktı birimleri çıktıların geçmiş değerlerinin bir fonksiyonudur. Başka bir ifade ile çıktının geçmiş değerleri ağırlıklandırılmaktadır ve ağırlıklandırma işleminde kullanılan parametreler için parametre paylaşımı söz konusudur (Goodfellow ve diğerleri, 2016, s.375).

3.3.1 RNN Mimarisi

Zamana göre sıralı bir veri ele alındığında standart RNN mimarisi, bir zaman indisi (t) yardımıyla ifade edilebilir. Standart bir RNN mimarisinde girdi katmanı, gizli katman ve çıktı katmanı olmak üzere katmanlar mevcuttur. RNN mimarisi gizli katmanda yer alan geçmiş bilginin dahil edildiği bir döngü yapısıyla standart bir sinir ağından farklılaşmaktadır. Standart bir RNN mimarisi Şekil 41 'de verilmiştir.



Şekil 41: Yinelemeli Sinir Ağı Döngüsü ve Açılmış Gösterimi

Kaynak: Weijang Feng, Naiyang Guan, Yuan Li, Xiang Zhang ve Zhigang Luo, “Audio Visual Speech Recognition With Multimodal Recurrent Neural Networks”. In *2017 International Joint Conference on neural networks (IJCNN)*, 2017, May, s. 681-688, IEEE.

Şekil 41’de x girdi katmanını, h gizli katmanı ve o çıktı katmanını göstermektedir. Şeklin sol tarafında yer alan kısım, gizli katmanda gerçekleşen döngüyü (loop) içermektedir. Standart bir RNN mimarisinde bu döngü birçok kez tekrar edilmektedir. Sağ tarafta ise bu döngünün açılmış (unfolded) hali yer almaktadır. Burada t zamanında $h^{(t)}$ gizli katmanı $x^{(t)}$ girdisi ile birlikte bir önceki dönemden elde edilen $h^{(t-1)}$ bilgisinin ağırlıklandırılmasıyla hesaplanmaktadır. Bu katmandan $o^{(t)}$ çıktısı elde edilmektedir. Burada U , W ve V ağırlıklandırma parametrelerini ifade etmektedir ve bu parametreler döngü boyunca paylaşılmaktadır. Şekil 41’de yer alan standart RNN mimarisinde yer alan döngünün genel gösterimi matematiksel olarak,

$$a^{(t)} = b_1 + Wh^{(t-1)} + Ux^{(t)}$$

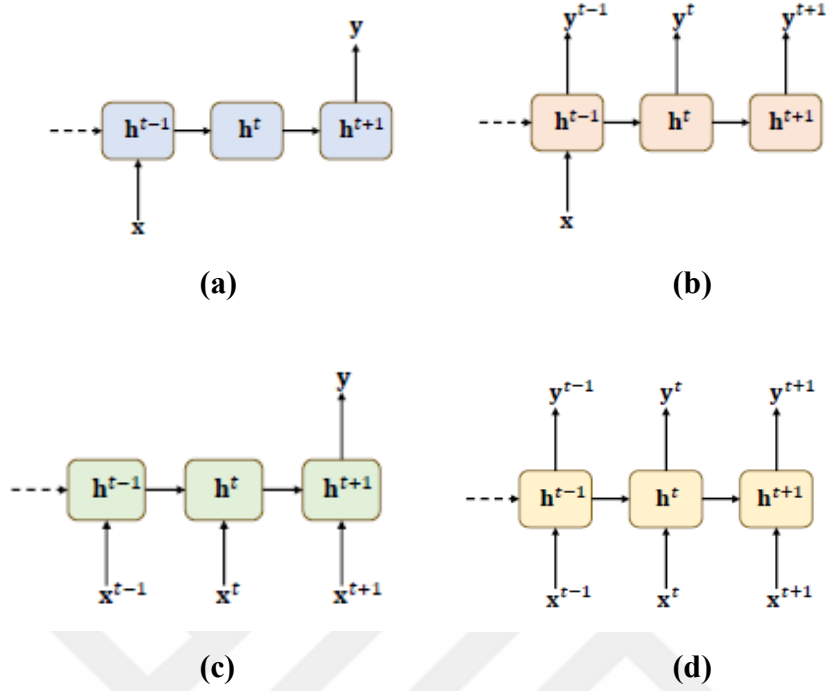
$$h^{(t)} = \sigma(a^{(t)})$$

$$o^{(t)} = b_2 + Vh^{(t)}$$

$$\hat{y}^{(t)} = \text{softmax}(o^{(t)})$$

şeklinde ifade edilebilir. Burada b_1 ve b_2 sapma vektörlerini göstermektedir. U matrisi girdi katmanından gizli katmana, W matrisi gizli katmandan gizli katmana ve V matrisi ise gizli katmandan çıktı katmanına geçişte kullanılan ağırlık matrisini ifade etmektedir. σ ile aktivasyon fonksiyonu temsil edilmektedir. Burada aktivasyon fonksiyonu olarak genellikle hiperbolik tanjant (tanh) ve sigmoid fonksiyonları tercih edilmektedir (Goodfellow ve diğerleri, 2016, s.380; Geron, 2017, s.381; Feng ve diğerleri, 2017). Ayrıca ReLU aktivasyon fonksiyonu da kullanılabilir (Talathi ve Vartak, 2016). Son aşamada gizli katmandan elde edilen çıktı birimine softmax aktivasyon fonksiyonu uygulanarak nihai çıktı elde edilmektedir. Her bir gizli katman bir önceki gizli katmandan gelen bilgileri içerdiğinden her t zamanındaki çıktı, başlangıçtan beri tüm girdilerin bilgisini kısmen içermektedir.

RNN algoritmasında yer alan girdi ve çıktıların sayısına göre çeşitli RNN mimarileri bulunmaktadır. Bu çeşitlilik RNN’nin farklı amaçlarla kullanımına imkan sağlamaktadır. Genel olarak dört farklı RNN mimarisi bulunmaktadır. Bunlar Şekil 42’de verilmiştir.



Şekil 42: Çeşitli RNN Mimarileri

Kaynak: Anu Jagannath, Jithin Jagannath ve Prem S. P. V. Kumar, “A Comprehensive Survey on Radio Frequency (rf) Fingerprinting: Traditional Approaches, Deep Learning, and Open Challenges”, 2022, *arXiv preprint arXiv:2201.00680*.

Şekil 42’de yer alan mimariler,

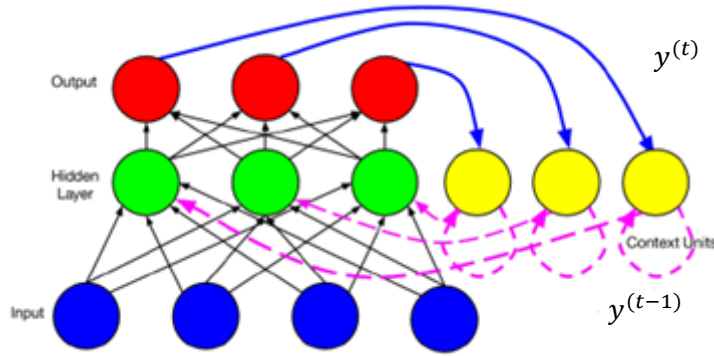
- (a) bir girdiden bir çıktıya (one-to-one)
- (b) bir girdiden çok çıktıya (one-to-many)
- (c) çok girdiden bir çıktıya (many-to-one)
- (d) çok girdiden çok çıktıya (many-to-many)

olarak adlandırılmaktadır. One-to-one mimarisi, görüntü sınıflandırma amacıyla kullanılmaktadır. One-to-many mimarisi, yazı içeren görüntüden sözcük çıkarımı veya müzik üretimi gibi alanlarda tercih edilebilir. Many-to-one mimarisinin duygu analizinde kullanımı yaygındır. Örneğin, olumlu ve olumsuz görüşleri tespit etmeye olanak sağlamaktadır. Many-to-many mimarisi otomatik yapılan tercüme işlemlerinde ve video kare etiketlemede kullanılmaya uygundur (Arif, 2020, s.33; Sharma ve Malik, 2021, s.589).

RNN mimarisinin ortaya çıktığı ilk zamanlarda çeşitli RNN mimarileri önerilmiştir. Bunlardan en bilinenleri Jordan ve Elman ağlarıdır.

3.3.1.1 Jordan Ağları

İlk özel RNN mimarisi türlerinden biri olan Jordan ağları, Jordan (1986) tarafından önerilmiştir. Jordan ağları, klasik ileri beslemeli sinir ağı yapısında olduğu gibi üç temel katmandan oluşmaktadır. Bu katmanlar girdi katmanı, gizli katman ve çıktı katmanıdır. Burada bir katmandaki tüm nöronlar bir sonraki katmandaki tüm nöronlarla bağlantılıdır. Ancak Jordan ağlarında ek olarak gizli katmana paralel yapıda bir bağlam katmanı bulunmaktadır. Bu bağlam katmanı, gecikme katmanı olarak da bilinmektedir (Lewis, 2016, s.95). Jordan ağlarının yapısı Şekil 43’de verilmiştir.



Şekil 43: Jordan Ağları

Kaynak: Zachary C. Lipton, John Berkowitz ve Charles Elkan, “A Critical Review of Recurrent Neural Networks for Sequence Learning”, 2015, *arXiv preprint arXiv:1506.00019*.

Şekil 43’de görüldüğü gibi, çıktı katmanının bir kopyası bağlam katmanına aktarılmaktadır. Ayrıca bağlam katmanının her birimi kendi içerisinde iç bağlantılara sahiptir. Çıktı katmanının kopyası bir sonraki zaman adımında yer alan gizli katmana bir girdi olarak eklenir. Böylece ağa bir hafıza özelliği kazandırılarak geçmiş bilgiler dikkate alınmaktadır (Lipton ve diğerleri, 2015). Jordan ağlarının matematiksel yapısı,

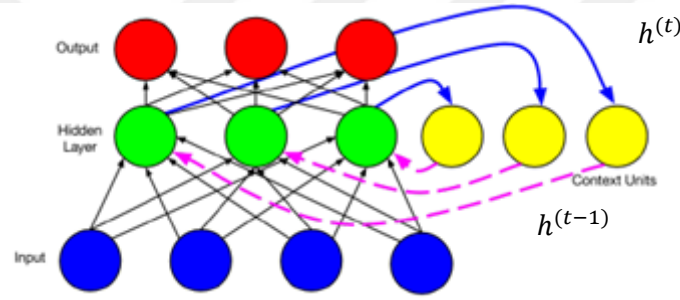
$$h^{(t)} = \sigma_1(b_1 + Wy^{(t-1)} + Ux^{(t)})$$

$$y^{(t)} = \sigma_2(b_2 + Vh^{(t)})$$

şeklinde ifade edilmektedir. Burada yer alan gösterimler RNN mimarisinde tanımlandığı şekilde kullanılmaktadır. σ_1 ve σ_2 uygun aktivasyon fonksiyonlarını göstermektedir (Skansi, 2018, s.141).

3.3.1.2 Elman Ağları

RNN mimarisi türlerinden biri olan Elman ağları, Elman (1990) tarafından önerilmiştir. Bu ağların yapısı Jordan ağları ile benzer olmakla birlikte temel fark, geçmiş bilgisinin çıktı katmanını yerine gizli katmandan alınmasıdır. Elman ağlarında bağlam katmanında yer alan nöronlar ile gizli katmandaki nöronlar birebir eşleşmektedir. Elman ağlarının yapısı Şekil 44’de verilmiştir (Lewis, 2016, s.83).



Şekil 44: Elman Ağları

Kaynak: Zachary C. Lipton, John Berkowitz ve Charles Elkan, “A Critical Review of Recurrent Neural Networks for Sequence Learning”, 2015, *arXiv preprint arXiv:1506.00019*.

Şekil 44’de görüldüğü gibi, burada Jordan ağlarından farklı olarak çıktı katmanını yerine gizli katmanın çıktısının bir kopyası bağlam katmanına aktarılmaktadır. Bu kopya bir sonraki zaman adımında yer alan gizli katmanın çıktısının hesaplanmasında kullanılmaktadır. Bir başka ifade ile her bir gizli katmanın gecikmeli değeri bu katmana ek bir girdi olarak alınmaktadır. Böylece bağlam katmanını yardımıyla ağda gecikmeleri dikkate alan bir hafıza yapısı elde edilmektedir (Cruse, 2006, s.146). Matematiksel olarak Elman ağlarının genel yapısı,

$$h^{(t)} = \sigma_1(b_1 + Wh^{(t-1)} + Ux^{(t)})$$

$$y^{(t)} = \sigma_2(b_2 + Vh^{(t)})$$

şeklinde ifade edilmektedir. Burada yer alan gösterimler RNN mimarisinde tanımlandığı şekilde kullanılmaktadır. σ_1 ve σ_2 uygun aktivasyon fonksiyonlarını göstermektedir (Wang ve diğerleri, 2021).

3.3.2 Yinelemeli Sinir Ağlarında Geri Yayılım

RNN algoritmasında eğitim aşaması klasik bir sinir ağındaki ile benzer biçimde yapılmaktadır. İleri beslemeli ağlarda ağı eğitiminde geri yayılım algoritması kullanılmaktadır. RNN de benzer olarak geri yayılım algoritmasını kullanmaktadır. Ancak buradaki yayılma işlemi zamana bağlı olmaktadır. Bu yayılma işleminde öncelikle zamana bağlı oluşan döngü açılımı yapılarak her adımı bir zaman dönemine denk gelen ileri beslemeli sinir ağı elde edilir. Elde edilen ileri beslemeli ağ geri yayılım algoritması ile eğitilir (Lewis, 2016, s.73). Geri yayılım algoritmasının bu şekilde zamana bağlı olarak gerçekleştirilmesine *zamana göre geri yayılım* (backpropagation through time, BPTT) denir. Hesaplama kolaylığı ve hızlı sonuç vermesinden dolayı BPTT yöntemi en çok kullanılan yöntemdir (Werbos, 1990).

Zamana göre geri yayılım, gradyanlar hesaplanırken ilgili kısmi türevlerin elde edilmesi için zincir kuralından faydalanmaktadır. RNN’de gizli katmanın çıktısı kendisinin zamana göre gecikmeli değerinin bir fonksiyonudur. Bu nedenle zincir kuralı zamana göre gecikmeler üzerinden uygulanır. Her bir gizli katmana ait gradyanlar hesaplanırken gizli katmanın ilgili zamandaki çıktısından elde edilen kayıp fonksiyonundan faydalanılmaktadır. Standart bir RNN mimarisinin gizli katmanı ve çıktısı ilgili parametrelerin birer fonksiyonu olarak,

$$h^{(t)} = f(x^{(t)}, h^{(t-1)}, w)$$

$$o^{(t)} = g(h^{(t)}, v)$$

şeklinde gösterilebilir. $o^{(t)}$ çıktısı için kayıp değeri, tüm zamanlara göre hesaplanan kayıp fonksiyonunun ortalaması alınarak elde edilir. Burada kayıp fonksiyonu olarak (l) genellikle negatif logaritmik olabilirlik fonksiyonu kullanılmaktadır. Bu kayıp fonksiyonu yardımıyla $y^{(t)}$ gerçek değeri için $o^{(t)}$ çıktısının kayıp değeri,

$$L(x^{(1)}, \dots, x^{(T)}, y^{(1)}, \dots, y^{(T)}, w, v) = \frac{1}{T} \sum_{t=1}^T l(y^{(t)}, o^{(t)})$$

olarak hesaplanır. Burada T toplam zaman adımlarının sayısını ifade etmektedir. Bu L fonksiyonu gradyan hesaplanırken minimize edilecek amaç fonksiyonudur. Bu amaç fonksiyonunun ağırlık parametrelerine göre kısmi türevleri alınarak gradyanlar hesaplanır. w parametresi için bu kısmi türev, zincir kuralı yardımıyla,

$$\begin{aligned} \frac{\partial L}{\partial w} &= \frac{1}{T} \sum_{t=1}^T \frac{\partial l(y^{(t)}, o^{(t)})}{\partial w} \\ &= \frac{1}{T} \sum_{t=1}^T \frac{\partial l(y^{(t)}, o^{(t)})}{\partial o^{(t)}} \frac{\partial g(h^{(t)}, v)}{\partial h^{(t)}} \frac{\partial h^{(t)}}{\partial w} \end{aligned}$$

olarak hesaplanır. Burada toplam sembolü içerisinde yer alan ilk iki terimin hesaplanmasında sorun yoktur. Ancak $\frac{\partial h^{(t)}}{\partial w}$ teriminde ağırlık parametresine göre kısmi türevi söz konusudur. Buradaki gizli katman $h^{(t)}$, kendi gecikmesinin ($h^{(t-1)}$) ve ağırlık parametresinin (w) bir fonksiyonudur. Aynı şekilde $h^{(t-1)}$ 'de w 'ya bağlıdır. Bu bilgiden hareketle $\frac{\partial h^{(t)}}{\partial w}$ türevi zincir kuralı yardımıyla,

$$\frac{\partial h^{(t)}}{\partial w} = \frac{\partial f(x^{(t)}, h^{(t-1)}, w)}{\partial w} + \frac{\partial f(x^{(t)}, h^{(t-1)}, w)}{\partial h^{(t-1)}} \frac{\partial h^{(t-1)}}{\partial w}$$

şeklinde hesaplanır. Benzer şekilde geriye doğru yineleme yapılarak gradyanların $t = 1$ 'e doğru zamana yayılımı elde edilmektedir. (Zhang ve diğerleri, 2021, s. 357; Goodfellow ve diğerleri, 2016, s.385).

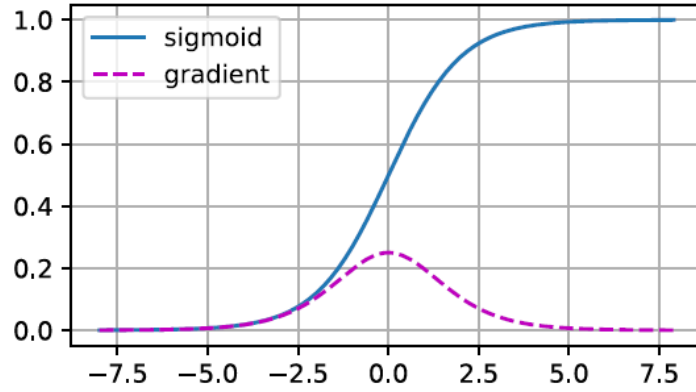
Zamana göre geri yayılımda t büyük olduğunda zincir kuralının hesaplanması uzun sürebilmektedir. Bu sorunun çözümü için zaman boyunca kesilmiş geri yayma (Jaeger, 2002) ve rassal kesme (Tallec ve Ollivier, 2017) gibi yöntemler önerilmiştir.

3.3.3 Gradyan Kaybı ve Patlaması

Zamana göre geri yayılım algoritmasının temel dezavantajlarından bir tanesi gradyan patlaması ve gradyan kaybıdır. Gradyanlar, ağırlık parametreleri hesaplanırken kullanılmakta ve zamana göre çarpımsal olarak bağlı olan kısmi türevlerden oluşmaktadır. Bu türevlerin adımlar ilerledikçe çok hızlı büyümesi veya küçülerek kaybolması ağırlık öğrenmesi için tehlike oluşturmaktadır. Gradyanların hızlı büyümesi durumuna literatürde *gradyan patlaması* (exploding gradient) adı verilirken; gradyanların azalarak sifıra yaklaşması durumuna *gradyan kaybı* (vanishing gradient) adı verilmektedir (Zhang ve diğerleri, 2021, s.184).

Gradyan patlaması durumunda ağırlık çok büyük değerler üreterek doğru sonuçtan uzaklaşma riski taşıyacaktır. Bu soruna pratik bir çözüm olarak, belirlenen bir eşik değeri yardımıyla çok fazla büyüyen gradyanların kesilmesi ve böylece patlamanın önüne geçilmesi önerilmiştir. Bu nedenle gradyan kaybı, gradyan patlamasına göre daha önemli bir problemdir. Bu problem Bengio ve diğerleri (1994) ve Hochreiter (1998) çalışmalarında detaylı olarak ele alınmıştır. Gradyan kaybı durumunda güncel ağırlık parametreleri çok fazla değişmeden kalacağından optimum çözüme yaklaşmak zorlaşacaktır (Geron, 2017, s.275). Ayrıca gradyan kaybından ötürü zaman dönemi ilerledikçe ağırlık ilk girdilere karşı hassasiyeti azalacak ve başlangıçtan getirilen bilgi gittikçe daha az dikkate alınmaya başlanacaktır (Graves, 2012, s.38).

Gradyan kaybı problemini önlemek için dikkat edilmesi gereken noktalardan biri ağırlık parametrelerine ait başlangıç değerlerinin uygun seçimidir. Gradyanlar ile bunlara bağlı ağırlık parametrelerinin kontrol altında tutulabilmesi için başlangıç parametrelerinin seçimine özen gösterilmelidir. Gradyan kaybının diğer bir önlemi uygun aktivasyon fonksiyonu seçimidir. Örneğin sıklıkla kullanılan sigmoid fonksiyonunun gradyanı küçük veya büyük girdi değerlerinde kaybolmaktadır. Bu durum Şekil 45’de gösterilmiştir.



Şekil 45: Sigmoid Fonksiyonu ve Gradyanı

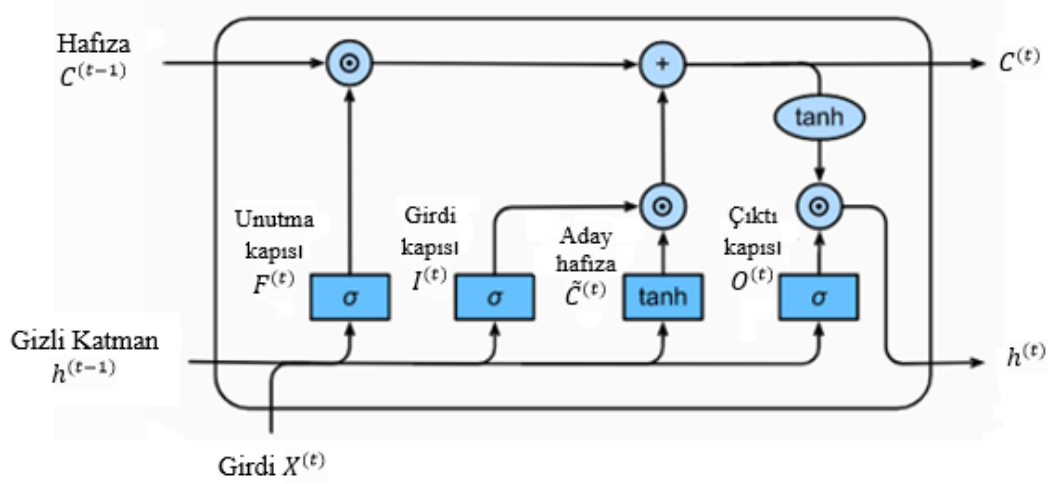
Kaynak: Aston Zhang, Zachary C. Lipton, Mu Li ve Alexander J. Smola, “Dive Into Deep Learning”, 2021, *arXiv preprint arXiv:2106.11342*.

Bu durumda geri yayılım sürecinde çarpımsal gradyanların kaybolması riski yüksektir. Buna çözüm olarak sigmoid fonksiyonu yerine türevi 0 ve 1 değerlerini alan ReLU aktivasyon fonksiyonunun kullanımı önerilmektedir (Zhang ve diğerleri, 2021, s.185).

Gradyan kaybının önlenmesi için en sık tercih edilen çözüm ise standart RNN yerine Uzun-Kısa Süreli Bellek Ağları (Long-short Term Memory Networks, LSTM) kullanılmasıdır (Aggarwal, 2018, s.289).

3.3.4 Uzun - Kısa Süreli Hafıza Ağları (LSTM)

RNN mimarisinin temel problemlerinden biri gradyan kaybı ve patlamasıdır. Diğer bir problem ise uzun geçmişten gelen bilgiyi ağı saklayamamasıdır. Çünkü RNN kısa süreli bir belleğe sahiptir. Bu problem sebebiyle özellikle büyük verilerde kullanımı dezavantaj oluşturmaktadır. Bu problemlerin çözümü olarak Hochreiter ve Schmidhuber (1997) tarafından Uzun-Kısa Süreli Hafıza Ağları (LSTM) önerilmiştir. Bu ağı klasik RNN’den ayıran temel özellik, geçmişten gelen önemli bilgileri dahil eden kapıların (geçitlerin) bulunmasıdır. Bu kapılar sayesinde uzun geçmişten gelen bilgiler ayıklanarak gerekli olanlar hafızaya eklenmekte diğerleri ise elimine edilmektedir. Şekil 46’da bir LSTM mimarisinde yer alan bir hücrede gerçekleşen akış diyagramı verilmiştir.



Şekil 46: LSTM Hücresi

Kaynak: Aston Zhang, Zachary C. Lipton, Mu Li ve Alexander J. Smola, “Dive Into Deep Learning”, 2021, *arXiv preprint arXiv:2106.11342*.

Şekil 46’da görüldüğü gibi LSTM’de hücreye girdi olarak alınan t zamanına ait girdi ($X^{(t)}$) ve bir önceki zaman dönemine ait gizli katman bilgisi ($h^{(t-1)}$) farklı kapılardan geçerek işlenmektedir. Ayrıca bu mimariye özgü olarak bir önceki zaman diliminden elde edilen hafıza bilgisi ($C^{(t-1)}$) ilgili zaman diliminde hücreye girdi olarak alınmaktadır.

Hücre içerisinde girdilerin işlendiği kapılar sırasıyla unutma kapısı ($F^{(t)}$), girdi kapısı ($I^{(t)}$) ve çıktı kapısıdır ($O^{(t)}$). *Unutma kapısı* uzun süreli geçmişten gelen ve bir önceki hücreden aktarılan hafızada yer alan bilgilerin ne kadarının tutulup ne kadarının unutulacağını kontrol eder. Unutma kapısının t zamanındaki hesaplanması,

$$F^{(t)} = \sigma(U_f X^{(t)} + W_f h^{(t-1)} + b_f)$$

şeklinde dir. Burada aktivasyon fonksiyonu olarak (σ) sigmoid fonksiyonu kullanılmaktadır. Böylece kapının değerlerinin (0,1) aralığında yer alması sağlanmaktadır. Bu aktivasyon fonksiyonunun kullanımı girdi ve çıktı kapıları için de aynı şekilde gerçekleşmektedir. Burada ve ilerleyen adımlarda yer alan U ve W ilgili kapıya veya hücreye ait ağırlıklandırma parametreleridir.

Girdi kapısı yeni gelen verilerin ne ölçüde dikkate alınacağını ve uzun vadeli hafızaya ekleneceğini belirlemektedir. Girdi kapısının değeri,

$$I^{(t)} = \sigma(U_i X^{(t)} + W_i h^{(t-1)} + b_i)$$

şeklinde hesaplanmaktadır. Bu noktada kapılara ek olarak aday hafıza (bellek) hücresinin ($\tilde{C}^{(t)}$) hesaplanması gerekmektedir. Bu hücre,

$$\tilde{C}^{(t)} = \tanh(U_c X^{(t)} + W_c h^{(t-1)} + b_c)$$

şeklinde hesaplanmaktadır. Burada kapılardan farklı olarak sigmoid fonksiyonu yerine $\tanh$ aktivasyon fonksiyonu kullanılmaktadır. Son kapı olan *çıkı kapısı* ise hangi bilgilerin çıkı olarak alınacağını kontrol eder. Bu kapının değeri

$$O^{(t)} = \sigma(U_o X^{(t)} + W_o h^{(t-1)} + b_o)$$

olarak hesaplanır. Kapıların değerleri hesaplandıktan sonra hafıza hücresinin ($C^{(t)}$) güncel değerini elde etmek için unutma kapısı ile önceki döneme ait hafıza hücresi bilgisi ve girdi kapısı ile yeni verilerin bilgisi birleştirilir. Böylece yeni verilerden uygun olanlarının uzun dönem hafızaya eklenmesi ve hafızada yer alan önemsiz bilgilerin unutulması sağlanır. Bu işlem matematiksel olarak,

$$C^{(t)} = F^{(t)} \odot C^{(t-1)} + I^{(t)} \odot \tilde{C}^{(t)}$$

şeklinde ifade edilebilir. Burada $\odot$ skaler çarpımı ifade etmektedir. Böylece t zamanına ait hafıza değeri elde edilmiş olur ve bir sonraki hücreye girdi olarak aktarılır. Gizli birimin ve hücrenin çıktısının hesaplanması için ise bu hafıza değeri ile çıkı kapısının değerinden faydalanır. Gizli birimin ($h^{(t)}$) değeri,

$$h^{(t)} = O^{(t)} \odot \tanh(C^{(t)})$$

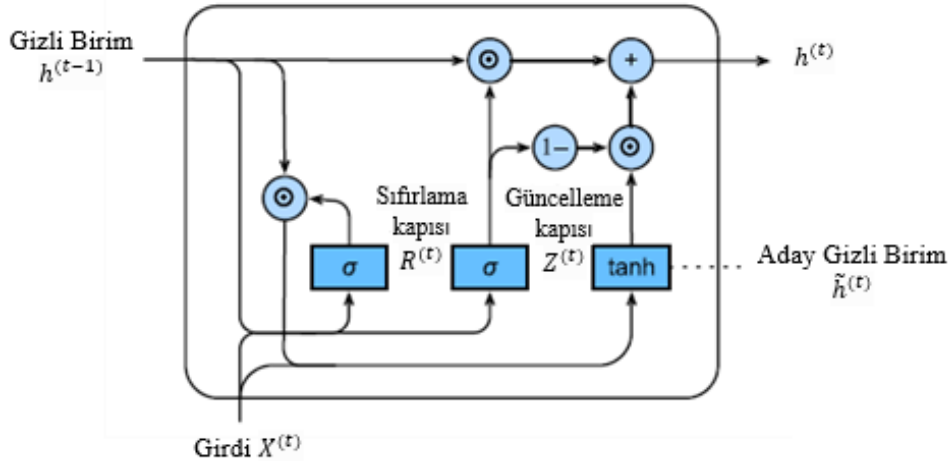
olarak elde edilir. Bu değer aynı zamanda hücreye ait genel çıkı değeridir. Nihai çıkı olarak,

$$h^{(t)} = y^{(t)}$$

olarak alınmaktadır. Bu $h^{(t)}$ değeri, $C^{(t)}$ ile beraber bir sonraki zaman dönemine girdi olarak aktarılır (Skansi, 2018, s.147; Geron, 2017, s.401; Graves, 2012, s.37; Siami-Namini ve diğerleri, 2019; Oualil ve diğerleri, 2017).

3.3.5 Geçitli Yinelemeli Birimler (GRU)

RNN mimarisinin uygulanması esnasında ortaya çıkabilecek gradyan kaybolması sorununa çözüm olarak sıklıkla kullanılan yöntemlerden bir diğeri de Geçitli Yinelemeli Birimlerdir (Gated Recurrent Units, GRU). Bu yöntem ilk olarak Cho ve diğerleri (2014) tarafından önerilmiştir. GRU, LSTM mimarisinin basitleştirilmiş özel bir halidir. Ancak daha az işlem yükü gerektirdiğinden çoğunlukla LSTM'den daha hızlı ve etkili bir performansa sahiptir (Chung ve diğerleri, 2014). Bu mimarinin LSTM'den farkı, unutma, girdi ve çıktı kapıları yerine sıfırlama (reset) ve güncelleme (update) kapılarının mevcut olması ve ek bir hafıza birimine ihtiyaç duyulmamasıdır. LSTM'de gizli katmanda değişen bilgi miktarı, unutma ve çıktı kapıları yardımıyla kontrol edilmektedir. GRU'da ise bu amaçla sadece sıfırlama kapısı kullanılmaktadır. Ancak genel olarak gizli birimdeki bilginin işlenmesi açısından iki mimari benzer bir yapıya sahiptir (Aggarwal, 2018, s.295). GRU mimarisinde t zamanında bir hücrede gerçekleşen süreç Şekil 47'de yer almaktadır.



Şekil 47: GRU Hücresi

Kaynak: Aston Zhang, Zachary C. Lipton, Mu Li ve Alexander J. Smola, “Dive Into Deep Learning”, 2021, *arXiv preprint arXiv:2106.11342*.

Şekil 47’de görüldüğü gibi, GRU hücresinde girdi olarak ilgili zaman dönemine ait bilgi ($X^{(t)}$) ve bir önceki zaman dönemine ait gizli katman bilgisi ($h^{(t-1)}$) alınmaktadır. Bu girdi sıfırlama ($R^{(t)}$) ve güncelleme ($Z^{(t)}$) kapıları boyunca işlenmektedir. *Sıfırlama kapısı*, geçmişten gelen bilgilerin ne kadarının hatırlanıp ne kadarının unutulacağına karar vermekte kullanılmaktadır. Bu kapının değeri,

$$R^{(t)} = \sigma(U_r X^{(t)} + W_r h^{(t-1)} + b_r)$$

şeklinde hesaplanmaktadır. Girdi verilerinin işlendiği diğer kapı ise *güncelleme kapısı*dır. Bu kapı, geçmişten gelen bilgilerin ne kadarının bir sonraki aşamaya kopyalanacağını kontrol edilmesini sağlamaktadır. Bu kapının değeri,

$$Z^{(t)} = \sigma(U_z X^{(t)} + W_z h^{(t-1)} + b_z)$$

olarak elde edilmektedir. Burada her iki kapı için de aktivasyon fonksiyonu (σ) olarak sigmoid fonksiyonu tercih edilmektedir. Böylece kapıların değeri (0,1) aralığında yer almaktadır. Bu değerler hesaplandıktan sonra öncelikle sıfırlama kapısının değeri kullanılarak, ilgili zaman dönemi için aday bir gizli birim ($\tilde{h}^{(t)}$) elde edilmektedir. Bu gizli birimin değeri,

$$\tilde{h}^{(t)} = \tanh(U_h X^{(t)} + (R^{(t)} \odot h^{(t-1)})W_h + b_h)$$

şeklinde hesaplanmaktadır. Buradan elde edilen değer (-1,1) aralığında yer alması için *tanh* aktivasyon fonksiyonu kullanılmaktadır. Bu noktada dikkat çeken durum, LSTM’de yer alan hafıza hücresinin GRU’da yer almaması ve hafıza hücresinin yerine getirdiği işlevin doğrudan aday gizli birim üzerinden gerçekleştirilmesidir.

Bir sonraki aşama, güncelleme kapısı kullanılarak aday gizli birim ($\tilde{h}^{(t)}$) ve geçmişten gelen gizli birimin ($h^{(t-1)}$) birleştirilmesidir. Böylece t zamanına ait nihai gizli birim değeri,

$$h^{(t)} = Z^{(t)} \odot h^{(t-1)} + (1 - Z^{(t)}) \odot \tilde{h}^{(t)}$$

olarak elde edilir. Bu noktada güncelleme kapısı ($Z^{(t)}$) geçmiş bilgiler ile yeni bilgiler arasında karar mekanizması olarak kullanılmaktadır. Güncelleme kapısının değeri 1’e

yaklařtıęında gemiřten gelen bilgilerin byk bir kısmı korunmuř olur. Buna karřılık bu deęer 0'a yaklařtıęında nihai olarak alınan gizli birim, aęırlıklı olarak ilgili zaman dnemine ait aday gizli birim tarafından belirlenir. GRU'da yer alan sıfırlama kapısı, genellikle kısa dnem iliřkileri tespit etmede etkili olurken, gncelleme kapısı ise uzun dnem iliřkileri tespit etmede etkilidir (Zhang ve dięerleri, 2021, s.367; Geron, 2017, s.404; Lewis, 2016, s.145).



4. BİST30 MUM FORMASYONU SINIFLANDIRMASI VE FİYAT HAREKETLİLİĞİ ÖNGÖRÜSÜ

Bu bölümde, daha önce anlatılan CNN, LSTM ve GRU algoritmaları kullanılarak hisse senedi fiyat hareketliliğine ilişkin sınıflandırma ve öngörü uygulamasına yer verilecektir. Bölümün devamında öncelikle uygulamanın teorik çerçevesini oluşturan teknik analize ilişkin temel kavramlar açıklanarak, literatürde derin öğrenme algoritmalarının bu alanda kullanıldığı çalışmalara değinilecektir. Ardından tezin uygulaması kapsamında verilerin oluşturulması ve düzenlenmesi aşamalarından bahsedilerek, mum formasyonları için sınıflandırma modeli ve fiyat hareketliliğinin incelenmesi için LSTM ve GRU modellerinin detayları sunulacaktır.

4.1 Teknik Analiz

Yatırımcıların, yatırım ortamlarını ve genel olarak piyasayı iyi tanımaları doğru yatırım kararları almaları açısından önem arz etmektedir. Sermaye piyasasının doğru takip edilerek iyi analiz edilmesi yatırımcının daha güvenli hareket biçimi belirlemesine olanak sağlamaktadır. Aksi durumda piyasalarda oluşabilecek beklenmeyen hareketlilikler, yatırımcının olası kâr fırsatlarını kaçırmasına ya da beklenmedik zararlar yaşamasına sebep olacaktır. Bu durumdan kaçınmak için, bireysel ve kurumsal olarak yapılacak yatırımlarda makro ve mikro düzeyde bilgilere ulaşmak önemlidir.

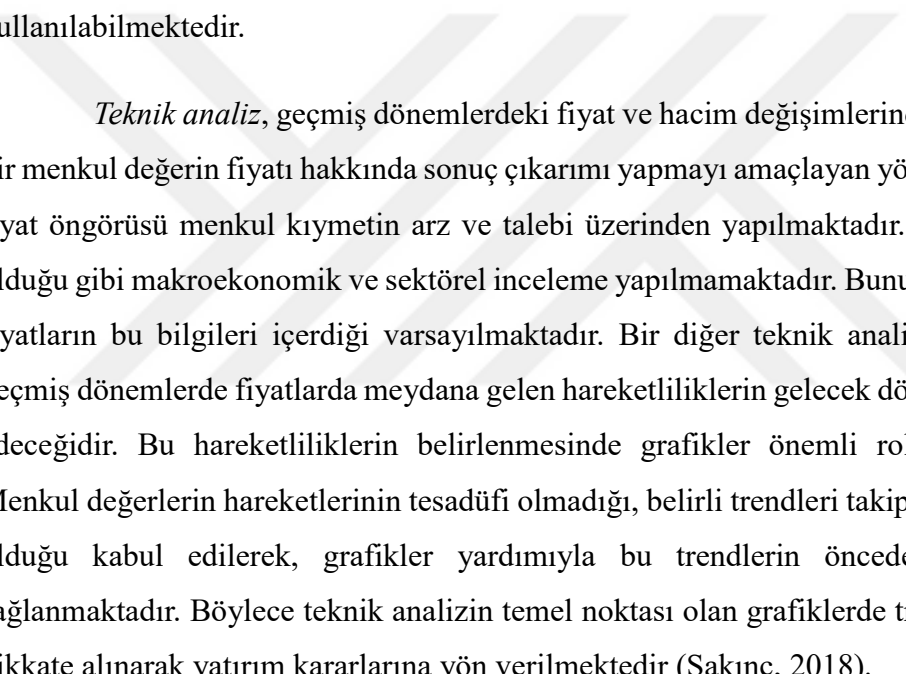
Piyasa analizindeki yatırımcıların bilgiye ulaşımı konusunu ele alan temel varsayımlardan biri Etkin Piyasa Hipotezi'dir (Fama, 1965). Bu hipoteze göre, menkul değerler piyasadaki tüm bilgiyi içermektedir ve yeni bir bilginin ortaya çıkması durumunda fiyatlar bu bilgiye göre kendini güncellemektedir. Dolayısıyla piyasanın etkin olduğu bir durumda yatırımcıların mevcut bilgileri kullanarak ortalama düzeyden daha fazla kâr elde etmesi mümkün olmayacaktır. Ancak gerçek hayatta piyasadaki tüm bilgilerin piyasada işlem yapan tüm kişi ya da kurumlara ulaşması kolay olmamaktadır. Bu durumda bilgilere ulaşan ve ulaşamayan kişiler arasında haksız rekabet durumu ortaya çıkacak ve asimetrik bilgi sorunu oluşacaktır (Çelik ve Taş, 2007; Bayraktar, 2012). Piyasanın tam etkin olmadığı durumlar için zayıf, yarı güçlü ve güçlü etkinlik kavramları ortaya çıkmıştır. Zayıf etkinlik durumunda, piyasa geçmiş fiyat bilgisini içermekte ancak

güncel durumunu anlık olarak yansıtamamaktadır. Dolayısıyla sadece geçmiş fiyat bilgisini kullanarak normalin üstünde bir getiri sağlanamayacaktır. Yarı güçlü etkinlik durumunda piyasa, geçmiş bilgisi ile beraber güncel olarak halka açıklanmış tüm bilgiyi de içermektedir. Bu durum kamuya açıklanmamış bilgilere ulaşan yatırımcıların piyasanın ortalama getirisinin üstünde bir kazanç sağlamasını mümkün kılmaktadır. Güçlü etkinlik durumunda ise, açıklanan ve açıklanmayan tüm bilgiler yatırımcılara eşanlı olarak ulaşmaktadır. Dolayısıyla çeşitli analizlerle piyasa ortalamasının üzerinde bir getiri elde etmek mümkün olmayacaktır (Fama, 1970).

Piyasa analizlerinde gerçek değerin ve gelecekteki değerin analiz edildiği temelde iki tür yöntem vardır. Bu yöntemler temel analiz ve teknik analizdir.

Temel analiz, menkul değerlerin nicel ve nitel veri analizi kullanılarak incelenmesi olarak ifade edilebilir. Temel analiz, genel ekonomik koşulların değerlendirildiği makroekonomik göstergelerle birlikte şirket bazındaki finansal durumların da değerlendirildiği analiz yöntemidir. Bu analiz yönteminde, tüm bilgiler göz önünde bulundurularak menkul değerin gerçek değeri ile ilgili değerlendirmeler yapılmaktadır. Dolayısıyla doğru zamanda doğru menkul değer ile işlem yapabilmek, gerçek değerin altında piyasa değerine sahip olan menkul değere yatırım yapılarak kazanç elde etmek hedeflenmektedir. Bu hedeflerin yanı sıra hem ülke ekonomisinin genel durumunu hem de sektör ve şirket bazlı değişimleri dikkate alarak bu değişimlerin fiyatlar üzerinde oluşturacağı etkileri öngörmek amaçlanmaktadır. Bu amaçlara ulaşabilmek için üç temel aşama gerçekleştirilmektedir. Bu aşamalar ekonomik analiz, sektör analizi ve şirket analizidir. *Ekonomik analiz*, menkul değerler genel ekonomik yapıdan etkilendiği için mevcut yapıyı oluşturan temel makroekonomik göstergelerle, ekonomik ve politik gidişatın incelenerek yatırım ortamına dair fikir elde edilmesidir. *Sektör analizi*, makroekonomik durumun sektörler üzerindeki etkisi farklı olduğu için sektör bazında inceleme yapılması ve uygun yatırım sektörünün belirlenmesidir. *Şirket analizi* ise, bir önceki aşama olan sektör analizinde belirlenen sektördeki yatırım yapılacak şirketin belirlenmesidir. Burada şirketin nitel ve nicel faktörleri değerlendirilir. Nitel olarak şirketin yönetim ve organizasyon yapısı, pazar payı gibi faktörler ele alınırken; nicel

olarak ise finansal rasyolardan hareketle performans değerlendirilmesi yapılmaktadır (Suresh, 2013; TSPAKB, 2011; Erdinç, 2004, s.48).

Menkul değerlerle ilgili fiyat araştırması yapılırken temel analiz oldukça zahmetli bir yöntemdir. Temel analizin başarılı olması, ekonomi ve finans alanlarında donanımlı kişiler tarafından yapılmasına bağlıdır. Temel analiz sonuçlarının değerlendirilmesi için de yine alanlarında yetkin kişilerin olması gerekmektedir. Menkul değerlerin fiyat araştırmasında kullanılan bir diğer yöntem de teknik analizdir. Bu iki yöntemin birbirlerine karşı bir üstünlüğü yoktur. İki yöntem bazı durumlarda farklı sonuçlar verse de başarılı sonuçlar elde etmek için birbirlerini tamamlayıcı olarak kullanılabilir. 

Teknik analiz, geçmiş dönemlerdeki fiyat ve hacim değişimlerinden yola çıkarak bir menkul değer fiyatı hakkında sonuç çıkarımı yapmayı amaçlayan yöntemdir. Burada fiyat öngörüsü menkul kıymetin arz ve talebi üzerinden yapılmaktadır. Temel analizde olduğu gibi makroekonomik ve sektörel inceleme yapılmamaktadır. Bunun yerine geçmiş fiyatların bu bilgileri içerdiği varsayılmaktadır. Bir diğer teknik analiz varsayımı ise geçmiş dönemlerde fiyatlarda meydana gelen hareketliliklerin gelecek dönemde de tekrar edeceğidir. Bu hareketliliklerin belirlenmesinde grafikler önemli rol oynamaktadır. Menkul değerlerin hareketlerinin tesadüfi olmadığı, belirli trendleri takip eden bir yapıda olduğu kabul edilerek, grafikler yardımıyla bu trendlerin önceden belirlenmesi sağlanmaktadır. Böylece teknik analizin temel noktası olan grafiklerde trend değişimleri dikkate alınarak yatırım kararlarına yön verilmektedir (Sakınç, 2018).

Teknik analizin temelini Dow Teorisi oluşturmaktadır. Dow Teorisi, 19. yüzyılın sonu ile 20. yüzyılın başında Charles Dow tarafından geliştirilen kurallar bütünüdür ifade etmektedir. Piyasa analizinin temel amacını fiyatların gidişatını belirlemek olduğunu ifade eden bu teori, modern teknik analizin altyapısını oluşturmaktadır. Dow teorisi altı ilkeye dayanmaktadır. Bu altı ilke şu şekildedir (Murphy, 1999, s.23; Korkmaz, 2013, s.160):

- 1) Ortalamalar tüm faktörleri yansıtmaktadır.

- 2) Piyasa birbirini kapsayan trendlerden oluşur. Piyasada temel olarak üç trend mevcuttur. Bu trendler; ana trendler, ikincil (ara) trendler ve dalgalanmalar (küçük trend) şeklindedir. Bir yıldan uzun süreli trende ana trend ya da birincil hareket denmekte, üç hafta ile üç ay arasında ise ikincil hareket denmekte, üç haftadan daha az olan trende ise dalgalanma olarak adlandırılmaktadır. Her trend kendinden bir büyük trendin düzeltilmesini temsil etmektedir.
- 3) Ana trendler üç aşama halinde ortaya çıkmaktadır. Bu aşamalar biriktirme, uyanış ve dağıtım dönemleridir. İlk aşama akıllı yatırımcı olarak adlandırılan küçük bir yatırımcı grubunun düşük fiyatlar geçerli iken biriktirme/toparlama yaptığı dönemdir. İkinci aşamada ekonomik şartlardaki uygunluktan kaynaklı olarak fiyatların yükselmeye başladığı trendi takip eden yatırımcıların da alıma geçtiği uyanış yani boğa piyasası dönemidir. Üçüncü aşama ise, yükselen piyasaları gören yatırımcıların satışa geçtikleri dağıtım dönemi yani ayı piyasası dönemidir.
- 4) Piyasa göstergeleri ile ortalamalar birbirleriyle örtüşmelidir.
- 5) Trend işlem hacmi ile desteklenmelidir.
- 6) Değişime dair sinyaller alınana kadar her trend geçerliliğini korumaktadır.

Teknik analiz kullanılarak yatırım biçimi belirleyen bir yatırımcı için pek çok yöntem mevcuttur. Dow teorisinde de yer alan trend analizi bunlardan biridir. Ayrıca destek ve direnç, formasyon ve indikatör analizleri alternatif analiz yöntemleridir. Bu analizlerin tek başına kullanımları yatırımcıya karar aşamasında fikir vermekle birlikte, bir arada kullanılmaları çok daha sağlıklı kararların ortaya çıkmasına olanak sağlamaktadır.

4.1.1 Destek ve Direnç

İlgilenilen yatırım aracının fiyat grafiğindeki hareketliliğin incelenmesi, destek ve direnç seviyelerinin tespitinde önemli rol oynamaktadır. Destek seviyesi, düşme eğiliminde olan fiyat düzeyinin yavaşladığı veya durduğu ve bir noktadan sonra yükselme eğilimine geçtiği seviyeyi ifade eder. Direnç seviyesi ise, desteğin aksine fiyattaki yükselme eğiliminin durarak düşme eğilimine geçtiği seviyeyi göstermektedir. Başka bir

ifade ile, geçmişte oluşan dip noktalar destek noktalarını oluştururken, tepe noktalar ise direnç noktalarını oluşturmaktadır. Bu noktaların tespiti fiyatların eğilimindeki değişimi göstermeleri açısından önemlidir. Destek ve dirençlerin ortaya çıkması arz talep dengesi ve yatırımcı psikolojisi ile doğrudan ilişkilidir. Yaklaşan destek noktası ile birlikte geçmiş destek noktalarını hatırlayan yatırımcı fiyatların yükselişe geçeceği düşüncesi ile alım yapmayı tercih ederek talep oluşturacaktır. Diğer taraftan yaklaşan direnç noktalarında fiyatın düşüş trendine geçeceğini öngören yatırımcılar satma eğilimine geçerek arz oluşturacaklardır (Suresh, 2013). Örnek bir hisse senedine ait destek ve direnç örneği Şekil 48’de yer almaktadır.



Şekil 48: Destek-Direnç Seviyeleri

Kaynak: Otabek Sattarov, Azamjon Muminov, Cheol W. Lee, Hyun K. Kang, Ryumduck Oh, Junho Ahn, Hyung J. Oh ve Heung S. Jeon, “Recommending Cryptocurrency Trading Points With Deep Reinforcement Learning Approach”, *Applied Sciences*. 10(4), 2020, 1506.

Hem destek hem direnç seviyelerinde genellikle yüksek işlem hacmi söz konusudur. Fiyatların destek noktasını kırması ile eski destek noktası direnç noktasına dönüşür. Benzer şekilde, fiyatların direnç noktasını kırması durumunda eski direnç noktası destek noktasına dönüşmektedir (Çetinyokuş ve Gökçen, 2002).

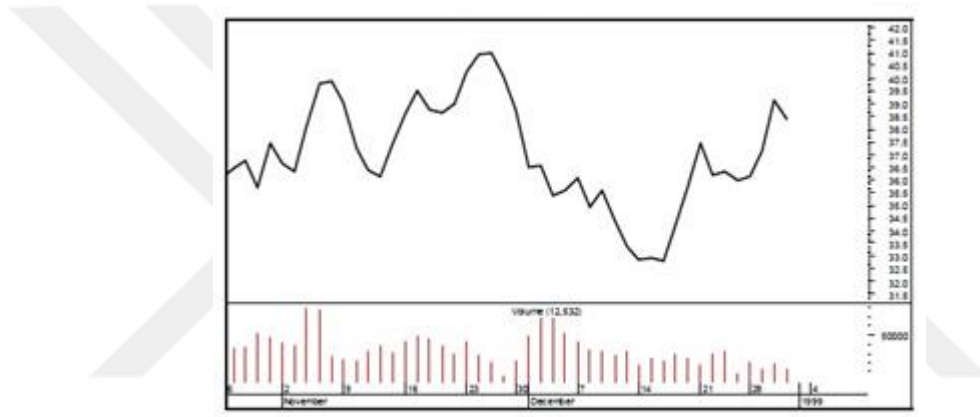
4.1.2 Grafikler

Teknik analizde fiyat hareketlerini görselleştirmek amacıyla grafiklerden sıklıkla faydalanılmaktadır. Çeşitli grafik türleri bulunmakla birlikte yaygın olarak tercih edilen grafik türleri çizgi grafiği, çubuk grafiği ve mum grafiğidir. Bunlar dışında X-O

grafığı (Çarpı sıfır grafığı, nokta şekil grafığı) ve kagi gibi grafik türleri de mevcuttur. Bu grafiklerde, dikey eksen fiyatları gösterirken, yatay eksen ilgilenilen döneme ait zaman bilgisini göstermektedir.

4.1.2.1 Çizgi Grafığı

Çizgi grafığı, piyasayı temsil eden tek bir değişkeni içeren en basit grafik türüdür. Hisse senetleri için kapanış değerleri kullanılarak, değerlerin bir doğru ile birleştirilmesiyle çizgi grafığı oluşturulmaktadır. Bir hisse senedine ait kapanış değerleri baz alınarak çizilen çizgi grafığı Şekil 49’da yer almaktadır.



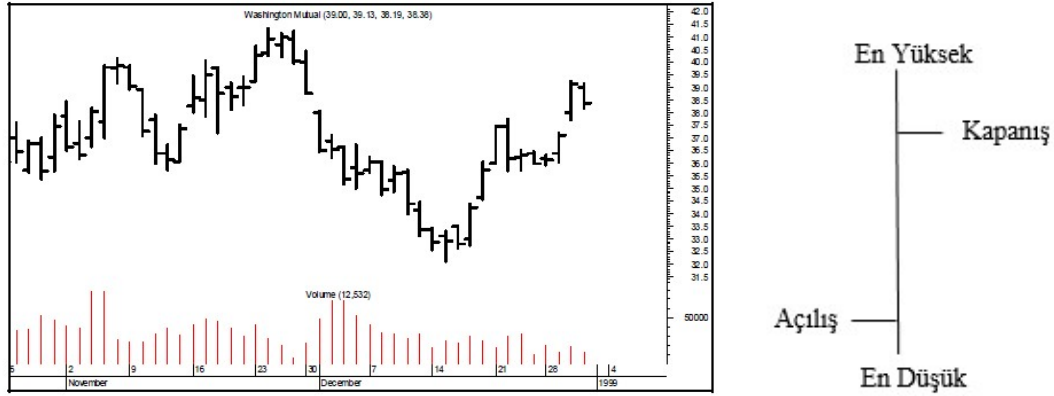
Şekil 49: Çizgi Grafığı

Kaynak: TSPAKB, “Analiz Yöntemleri Sermaye Piyasası Faaliyetleri İleri Düzey Lisansı Eğitimi”, Ankara: TSPAKB, 2011.

Günlük alınan hisse senetlerinde kapanış fiyatı önemli bilgiler içermekle birlikte, gün içinde ortaya çıkan fiyat dalgalanmalarını içermemektedir. Bu sebeple, çizgi grafiklerinin uzun vadeli analizlerde trendi incelemek amacıyla kullanılması daha uygun olmaktadır.

4.1.2.2 Çubuk Grafığı

Çubuk grafığı, çizgi grafığının verdiği bilgiye ek olarak her bir zaman dönemi içerisindeki fiyat değişimi bilgisini de göstermektedir. Günlük bir hisse senedine ait çubuk grafığında açılış ve kapanış fiyatları ile birlikte gün içerisindeki en yüksek ve en düşük fiyat bilgileri de yer almaktadır. Her bir gün için bu bilgileri içeren değerler bir çubukla temsil edilmektedir. Örnek bir çubuk grafığı Şekil 50’de verilmiştir.



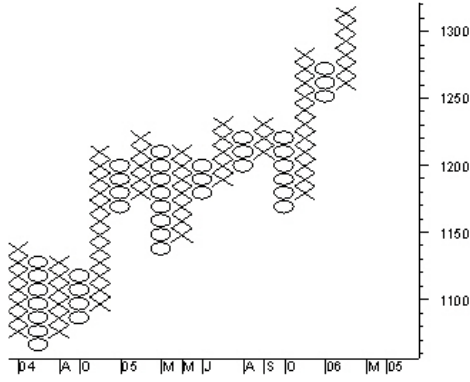
Şekil 50: Çubuk Grafiği

Kaynak: TSPAKB, “Analiz Yöntemleri Sermaye Piyasası Faaliyetleri İleri Düzey Lisansı Eğitimi”, Ankara: TSPAKB, 2011.

Şekil 50’de sol tarafta yer alan grafikte, belirlenen zaman aralığı için hisse senedinin fiyat hareketliliğine ait çubuk grafiği bulunurken, sağ tarafta ise bir çubuğun temsil ettiği değerler açık olarak ifade edilmiştir. Bir çubukta, belirlenen zaman dönemi (günlük, haftalık vb.) için fiyatların en düşük ve en yüksek değerleri çubuğun alt ve üst uç noktalarında gösterilmektedir. Ayrıca açılış ve kapanış fiyatları ise sırasıyla çubuğun soluna ve sağına konulan “-” işareti ile temsil edilmektedir. Kapanış fiyatı açılış fiyatından daha yüksek olabileceği gibi daha düşük de olabilir. Bazı durumlarda fiyatın yükseldiğini veya düştüğünü ifade etmek için farklı renkte çubuklar kullanılmaktadır (Di Lorenzo, 2013, s.7).

4.1.2.3 Nokta ve Şekil (X-O) Grafiği

Bu grafik, zaman ve işlem hacminin yer almaması sebebiyle günümüzde pek tercih edilmemektedir. X-O grafiğinde, araştırmacı tarafından belirlenen düzeyde fiyat hareketliliklerine odaklanılır. Dolayısıyla bu grafik türü önemli miktardaki artış ve azalışları göstermek üzerine tasarlanmıştır. Nokta ve şekil grafiğinin temsili bir gösterimi Şekil 51’de yer almaktadır.



Şekil 51: Nokta ve Şekil (X-O) Grafiği

Kaynak: A. S. Suresh, “A Study on Fundamental and Technical Analysis”, *International Journal of Marketing, Financial Services & Management Research*. 2(5), 2013, s.44-59.

Bu grafikte, araştırmacı tarafından belirlenen düzeyde fiyat artışı gerçekleştiğinde, bu artış “X” simgesi ile temsil edilirken, düşüş olduğunda da “O” ile temsil edilmektedir. Fiyatın yönünde değişim oldukça yan sütuna geçilerek artış ya da azalış işareti konulur (Suresh, 2013).

4.1.2.4 Kagi Grafiği

Kagi grafiği, nokta ve şekil grafiğine benzer bir yapıya sahiptir. Zaman ekseninin yer almadığı bu grafikte, fiyat değişimleri dikey çizgiler ile gösterilirken, bu çizgiler yatay çizgiler kullanılarak birleştirilir ve böylece sürekli bir grafik görüntüsü elde edilir. Örnek bir hisse senedine ait kagi grafiği Şekil 52’de yer almaktadır.



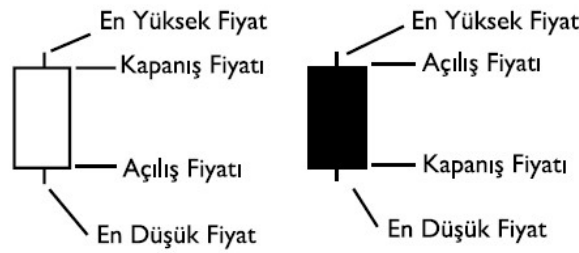
Şekil 52: Kagi Grafiği

Kaynak: TSPAKB, “Analiz Yöntemleri Sermaye Piyasası Faaliyetleri İleri Düzey Lisansı Eğitimi”, Ankara: TSPAKB, 2011.

Bu grafikte, genellikle kapanış fiyatı olan bir başlangıç fiyatından başlanarak fiyatın artışı durumunda yukarı, azalışı durumunda ise aşağı doğru dikey bir çizgi çizilerek ilerlenir. Araştırmacı tarafından önceden belirlenen düzeyde bir fiyat değişimi olduğunda yatay bir çizgi çizilerek grafik yön değiştirir ve ters yönde bir dikey çizgi çizilir. Fiyat yönünde belirlenen geri dönüş miktarında değişimler oldukça yatay çizgilerle bu değişimler gösterilir ve sürekli bir grafik elde edilir. Fiyatlar bir önceki en yüksek veya en düşük değeri aştığında, çizgi kalınlığı veya rengi değişir. Kalın veya koyu renkli çizgiler artış trendini ifade ederken, ince veya açık renkli çizgiler ise azalış trendini ifade etmektedir (Achelis, 2001, s.164).

4.1.2.5 Mum Grafiği

Bu grafik, 17. yüzyılda Japon pirinç piyasasındaki alım satım fiyatlarındaki değişimleri incelemek üzere Japonlar tarafından geliştirilmiş bir grafik türüdür. Mum grafikleri açılış, kapanış, günün en yüksek ve en düşük fiyatlarını içeren bir görsel sunması sebebiyle, teknik analizde kullanılan popüler grafik türlerinden biri olmuştur. Bir zaman dilimine ait mum 3 bileşenden oluşmaktadır. Bu bileşenler, mumun gövdesi, gölgesi (fitil) ve rengidir. Şekil 53’de klasik bir mum yapısı yer almaktadır.



Şekil 53: Mum Grafiğinin Temel Yapısı

Kaynak: Turhan Korkmaz, “Portföy Oluşturmada Temel ve Teknik Analizin Kullanılması”. *Portföy Yönetimi*, 2013, s. 146-177. M. Başar (Ed.). Eskişehir: Anadolu Üniversitesi.

Burada, açılış ve kapanış fiyatları arasındaki kısmı gösteren kutu şeklindeki yapı mumun gövdesidir. Bu gövdeyi dikey olarak kesen bir çizgi ile gösterilen kısım ise gölge veya fitil olarak adlandırılmaktadır. Bu gölgenin üst kısmı ilgili zaman periyodu (günlük, haftalık vb.) içindeki en yüksek fiyatı gösterirken, alt kısmı ise en düşük fiyatı göstermektedir. Mumun gövdesinin rengi piyasa hareketinin değişim yönünü

göstermektedir. Kapanış fiyatı açılış fiyatından yüksekse gövdenin rengi beyaz ya da yeşil olurken, düşük olduğunda ise gövdenin rengi siyah ya da kırmızı olarak gösterilmektedir. Beyaz renkli mum talebin fazla olduğuna işaret ederken, siyah renkli mum ise arzın fazla olduğuna işaret etmektedir. Temsili bir mum grafiği Şekil 54’de yer almaktadır.



Şekil 54: Mum Grafiği

Kaynak: TSPAKB, “Analiz Yöntemleri Sermaye Piyasası Faaliyetleri İleri Düzey Lisansı Eğitimi”, Ankara: TSPAKB, 2011.

Her bir zaman periyodu bir mum ile temsil edilirken, ilgili dönem içinde yer alan tüm periyotlara ait mumlar birleştirilerek mum grafiği elde edilir (Achelis, 2001, s.164; Nison, 2003, s.9; Di Lorenzo, 2013, s.9).

4.1.3 Formasyonlar

Fiyat hareketleri incelenirken, zaman içerisinde birbirine benzer yapıda çeşitli karakteristikler oluşmaktadır. Formasyon olarak bilinen bu karakteristik şekiller, piyasanın genel gidişatı ve fiyatlardaki değişim için yatırımcıya fikir vermek amacıyla sıklıkla kullanılan teknik analiz araçlarıdır. Genel olarak iki tür formasyon grubu vardır. Bunlardan ilki değişimi gösterirken, diğer grup devamlılığı göstermektedir. Bu değişim ve devamlılık yapısı, uzun dönem için trend üzerinden incelenebildiği gibi kısa dönem için mum grafikleri üzerinden de incelenebilir.

4.1.3.1 Trend Formasyonları

Trend formasyonlarında, yükseliş trendinden düşüşe ya da düşüş trendinden yükselişe geçişleri temsil eden formasyonlar değişim formasyonları olarak adlandırılır. Var olan trendin devam edeceğini gösteren formasyonlar ise devamlılık formasyonları olarak adlandırılmaktadır. Önemli değişim formasyonları arasında omuz-baş-omuz ve ters omuz-baş-omuz formasyonları başta olmak üzere, çift tepe/çift dip ve çoklu tepe/çoklu dip formasyonları yaygın olarak kullanılmaktadır. Önemli devamlılık formasyonları arasında ise üçgen formasyonları sıklıkla kullanılmaktadır. Üçgen formasyonları kendi içinde simetrik üçgen, yükselen üçgen ve düşen üçgen olarak adlandırılmaktadır. Bu formasyonlara ek olarak flama-bayrak, takoz-kama ve dikdörtgen-işlem bandı formasyonları kullanılmaktadır (Di Lorenzo, 2013, s.123).

4.1.3.2 Mum Formasyonları

Trend formasyonlarına benzer olarak mum grafikleri üzerinden de fiyatlar için değişim ve devamlılık yapısı incelenebilir. Tek bir mum veya birden fazla mumun bir araya gelerek ortaya çıkardığı yapılar, fiyat hareketi hakkında fikir veren formasyonlar oluşturur. Bu formasyonlardan değişimi gösterenler arasında, çekiç/asılı adam, yukarı/aşağı yutan, yukarı/aşağı harami, sabah/akşam yıldızı, üç yıldız yukarı/aşağı, üç içeride yukarı/aşağı, üç dıştan yukarı/aşağı formasyonları öne çıkmaktadır. Devam formasyonları arasında ise yükselen/düşen üçler sıklıkla kullanılmaktadır. Bu kısımda yer verilen görsellerdeki beyaz mumlar kapanış fiyatlarının açılış fiyatlarından daha yüksek olduğu günleri, siyah mumlar ise kapanış fiyatlarının açılış fiyatlarından daha düşük olduğu günleri belirtmektedir.

- **Çekiç ve Asılı Adam**

Çekiç formasyonu en dip seviyede ortaya çıkan düşüş trendinden yükseliş trendine geçişi gösteren, tek mumdan oluşan bir formasyondur. Kısa bir mum gövdesinin altında uzun bir gölgenin yer aldığı mum yapısıdır. Asılı adam ise çekiç formasyonunun tersine yükseliş trendinden düşüş trendine geçişi gösteren formasyondur. Şekil 55’de bu formasyonlar yer almaktadır.

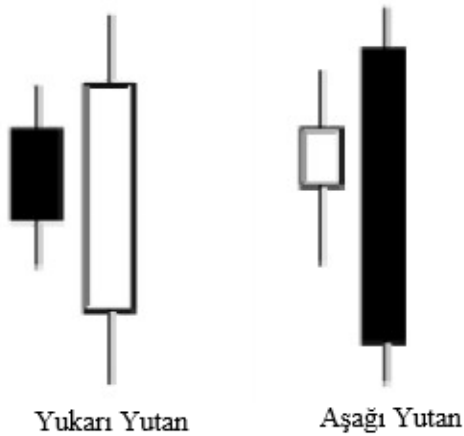


Şekil 55: Çekiç ve Asılı Adam Mum Formasyonları

Burada beyaz kutu çekiç formasyonunu, siyah kutu ise asılı adam formasyonunu göstermektedir (Di Lorenzo, 2013, s.138).

- ***Yukarı / Aşağı Yutan***

Yukarı yutan formasyon, büyük beyaz gövdeli mumdan önce yer alan küçük siyah gövdeli mumdan oluşan çift mumlu formasyondur. Aşağı yutan formasyon ise, büyük siyah gövdeli mumdan önce yer alan küçük beyaz gövdeli mumdan oluşan çift mumlu formasyondur. Her ikisinde de önemli olan görüntü, büyük gövdeli mumun, kendinden önceki küçük gövdeli mumu kapsayacak şekilde olmasıdır. Şekil 56’da bu formasyonlar yer almaktadır.

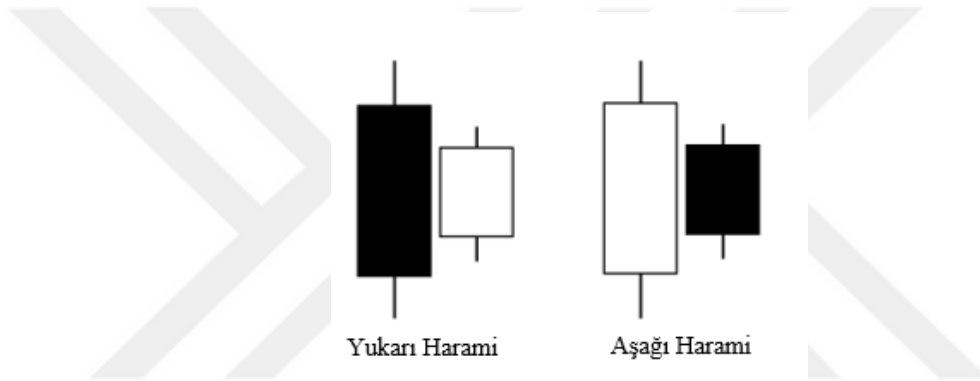


Şekil 56: Yukarı Yutan ve Aşağı Yutan Mum Formasyonları

Yukarı yutan formasyon, önemli bir düşüş trendinden sonra gelen güçlü bir yükseliş trendine işaret ederken, aşağı yutan formasyon ise önemli bir yükseliş trendinden sonra gelen güçlü bir düşüş trendine işaret etmektedir (Achelis, 2001, s.79).

- ***Yukarı / Aşağı Harami***

Harami formasyonu, büyük gövdeli bir mumdan sonra gelen ve kendinden önceki mum gövdesinin içinde kalan küçük gövdeli bir mumdan oluşan formasyondur. Hamile görüntüsünü andırdığı için hamile formasyonu olarak da adlandırılır. Yukarı ve aşağı harami formasyonları Şekil 57’de verilmiştir.



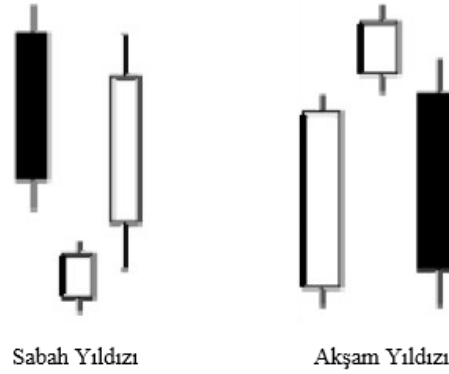
Şekil 57: Yukarı Harami ve Aşağı Harami Mum Formasyonları

Yukarı harami formasyonu, bir düşüş trendinin ardından ortaya çıkar ve trenddeki zayıflamayı temsil etmektedir. Siyah büyük gövdeli mumun içinde kalan küçük gövdeli beyaz mumdan oluşan ikili mum formasyonudur. Aşağı harami formasyonu ise yukarı harami formasyonunun tam tersidir (Nison, 2003, s.82).

- ***Sabah / Akşam Yıldızı***

Sabah ve akşam yıldızı formasyonları, trend değişimini gösteren önemli formasyonlardandır. Sabah yıldızında, devam eden güçlü bir düşüş trendinin ardından ilk olarak yükselişi temsil eden ve yıldız adı verilen küçük gövdeli bir mum gelir. Bu yıldız, daha güçlü yükselişi temsil eden büyük gövdeli beyaz bir mum tarafından desteklenir. Akşam yıldızında ise, devam eden güçlü bir yükseliş trendinin ardından, ilk olarak düşüşü temsil eden ve yıldız adı verilen küçük gövdeli bir mum gelir. Bu yıldız, daha güçlü

düşüşü temsil eden büyük gövdeli siyah bir mum tarafından desteklenir. Sabah ve akşam yıldızı formasyonları Şekil 58’de verilmiştir.



Şekil 58: Sabah Yıldızı ve Akşam Yıldızı Mum Formasyonları

Şekil 58’de görüldüğü gibi, sabah yıldızı bir yükseliş dönüş formasyonudur. Akşam yıldızı ise düşüşe geçişi gösteren bir dönüş formasyonudur. Ayrıca ortada yer alan yıldız mumları her iki formasyonda da beyaz veya siyah olabilir (Morris, 2006, s.94).

- **Üç Yıldız Yukarı / Aşağı**

Üç yıldız formasyonu, ardışık üç dojiden meydana gelmektedir. Doji kapanış ve açılış fiyatı aynı olan ve gövdesi çizgi şeklinde gösterilen tekli bir mum formasyonudur. Üç yıldız formasyonu, mevcut olan yükseliş ya da düşüş trendinin tersine dönme eğilimine işaret etmektedir. Üç yıldız yukarı ve üç yıldız aşağı formasyonları Şekil 59’da gösterilmiştir.

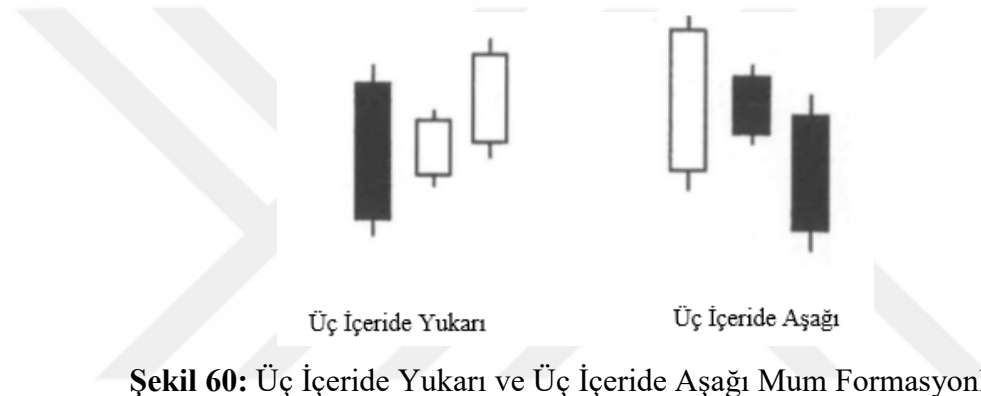


Şekil 59: Üç Yıldız Aşağı ve Üç Yıldız Yukarı Mum Formasyonları

Üç yıldız aşağı formasyonu, yükselen bir trendin tersine dönerek düşüşe geçişini gösterirken, üç yıldız yukarı formasyonu ise, düşen bir trendin tersine dönerek yükselişe geçişini göstermektedir (Bulkowski, 2008, s.812).

- **Üç İçeride Yukarı / Aşağı**

Üç içeride yukarı ve üç içeride aşağı formasyonları, Harami formasyonlarını doğrular nitelikte geliştirilmiş bir formasyondur. Ancak harami formasyonlarından farklı olarak üç mumun bir araya gelmesiyle oluşmaktadır. Üç içeride yukarı / aşağı mum formasyonlarına ait görsel Şekil 60’da verilmiştir.



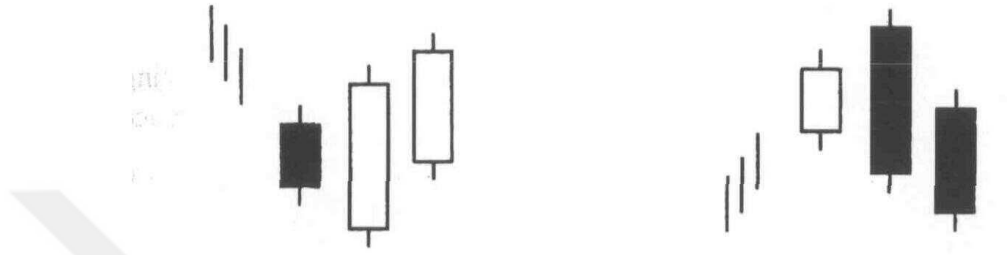
Şekil 60: Üç İçeride Yukarı ve Üç İçeride Aşağı Mum Formasyonları

Şekil 60’da görülen üç içeride yukarı formasyonunda düşüş trendini gösteren bir siyah gövdeli mumun ardından, bu mumun gövdesinin içerisinde kalacak şekilde gelen daha küçük gövdeye sahip beyaz gövdeli bir mum ve ardından bu mumun kapanışından daha yüksek bir kapanışa sahip bir diğer beyaz mumdan oluşmaktadır. Böylece yukarı harami formasyonundan anlaşılan düşüş trendindeki zayıflama ve yükseliş trendine doğru bir dönüş üçüncü bir beyaz gövdeli mum ile teyit edilmektedir.

Benzer şekilde üç içeride aşağı formasyonunda ise, yükseliş trendini gösteren beyaz gövdeli mumun ardından, bu mumun gövdesinin içerisinde kalacak şekilde gelen daha küçük gövdeye sahip siyah gövdeli bir mum ve ardından bu mumun kapanışından daha düşük bir kapanışa sahip bir diğer siyah mumdan oluşmaktadır. Böylece aşağı harami formasyonundan anlaşılan yükseliş trendindeki zayıflama ve düşüş trendine doğru bir dönüş üçüncü bir siyah gövdeli mum ile teyit edilmektedir (Morris, 2006, s.155).

- **Üç Dıştan Yukarı / Aşağı**

Bir önceki formasyonda açıklanan üç içeride yukarı / aşağı formasyonları ile harami formasyonları arasındaki ilişkiye benzer bir yapı burada da geçerlidir. Üç dıştan yukarı / aşağı formasyonları yutan formasyonların bir doğrulaması niteliğini taşımaktadır. Üç dıştan yukarı / aşağı mum formasyonlarına ait görsel Şekil 61’de verilmiştir.

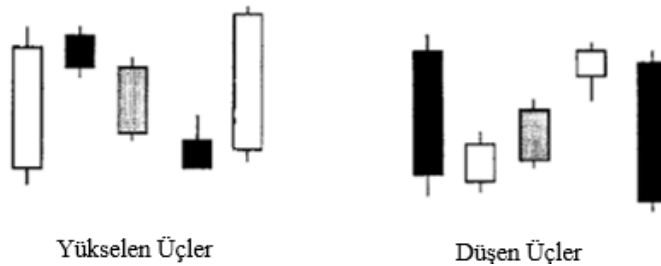


Şekil 61: Üç Dıştan Yukarı ve Üç Dıştan Aşağı Mum Formasyonları

Şekilde görüldüğü gibi, üç dıştan yukarı formasyonunda düşüş trendinden yükseliş trendine geçişi gösteren yukarı yutan formasyonu, daha yüksek kapanışa sahip beyaz gövdeli mum ile desteklenmektedir. Benzer olarak, üç dıştan aşağı formasyonunda ise, yükseliş trendinden düşüş trendine geçişi gösteren aşağı yutan formasyonu, daha düşük kapanışa sahip siyah gövdeli mum ile desteklenmektedir (Bulkowski, 2008, s.774).

- **Yükselen / Düşen Üçler**

Önceki formasyonlardan farklı olarak yükselen / düşen üçler trend devamlılık formasyonlarındandır. Yükselen üçler formasyonu devam eden bir yükseliş trendine işaret ederken, düşen üçler formasyonu ise devam eden bir düşüş trendine işaret etmektedir. Yükselen / düşen üçler mum formasyonlarına ait görsel Şekil 62’de verilmiştir.



Şekil 62: Yükselen Üçler ve Düşen Üçler Mum Formasyonları

Şekilde görülen yükselen üçler formasyonu, uzun gövdeli beyaz bir mum ile başlar. Ardından rengi önemli olmayan ancak daha küçük gövdeli birkaç mum ile devam eder ve sonrasında gelen büyük beyaz gövdeli bir mum yükseliş trendinin devamının geleceği sinyallerini verir. Düşen üçler formasyonunda ise, benzer süreç düşüş trendini gösteren iki büyük gövdeli siyah mum arasında kalan birkaç küçük gövdeli mum ile ortaya çıkmaktadır (Morris, 2006, s.262).

4.2 Literatür Taraması

Derin öğrenme yöntemleri, sosyal bilimlerde farklı alanlarda kullanılmaktadır. Ekonomi ve finans alanı, derin öğrenme yöntemlerinin sıklıkla uygulandığı sosyal bilim alanlarından bir tanesidir. Finansal varlıkların hareketlerine dair çıkarımlarda bulunmak ve öngörüler yapmak amaçlandığında, genellikle zaman serisi analizi gibi klasik yöntemler kullanılmaktadır. Bununla birlikte, son dönemde derin öğrenme yöntemleri aynı amaçla kullanılan klasik yöntemlere geniş bir alternatif oluşturmıştır. Finansal verilerin sayısal olarak analiz edilmesi için alternatif yöntemler ayrı ayrı veya bir arada kullanılabilir. Ancak finansal veriler sayısal olarak ifade edilebildiği gibi, mum grafikleri ile görsel olarak da ifade edilebilir. Derin öğrenme yöntemleri, sayısal verilerle birlikte görsel verileri analiz edebilme avantajına sahiptir. Literatürde, hisse senetleri gibi finansal varlıkların mum grafikleri kullanılarak analiz edilmesi ve ileriye yönelik finansal yatırım fikri oluşturulması amacıyla derin öğrenme yöntemlerinin kullanıldığı pek çok çalışma mevcuttur. Bu çalışmalarda CNN, LSTM, GRU gibi temel yöntemler başta olmak üzere alternatif derin öğrenme yöntemleri farklı yaklaşımlarla kullanılmaktadır. Genelleme yapılmaksızın, bu çalışmalar dört kapsamda incelenebilir. Bunlar; mum grafiklerinden formasyon tahmini, hisse senetleri için fiyat veya fiyat hareketliliği tahmini, hisse senedi için yatırım kararı olarak alma, satma veya bekletme tercihi ve daha geniş kapsamlı olarak finansal varlıklar için ticaret stratejisi veya portföy yönetimidir. Burada, özellikle mum grafiklerini analiz etmek için derin öğrenme yöntemlerinin bu kapsamlarda kullanıldığı son döneme ait literatür örneklerine yer verilecektir.

Derin öğrenme yöntemlerinin mum grafiklerini analiz etmek için kullanılmasındaki temel amaçlardan bir tanesi, mum grafiklerinde yer alan formasyonların doğru tespit edilmesi veya doğru sınıflandırılmasıdır. Çünkü bu

formasyonlar hisse senedi fiyatlarının gelecek dönemdeki hareketleri hakkında bilgi içermektedir ve yatırımcılara karar alma konusunda destekleyici bir fikir sağlamaktadır. Burada yer alan yaklaşım derin öğrenme açısından bakıldığında görüntü sınıflandırma (image classification) kapsamında ele alınabilir. Literatürde sık kullanılan formasyonların tespiti ve sınıflandırılması amacıyla çeşitli derin öğrenme yöntemlerinin kullanıldığı çalışmalar yer almaktadır (Velay ve Daniel, 2018; Chen ve Tsai, 2020, 2022; Chen ve diğerleri, 2020; Chen ve diğerleri, 2022). Velay ve Daniel (2018), Google’a ait C sınıfı hissesinde sık kullanılan iki formasyonun tespiti için CNN ve LSTM yöntemlerini kullanmış ve LSTM yönteminin daha başarılı olduğunu ortaya koymuşlardır. Chen ve Tsai (2020), zaman serilerinin farklı görüntü tiplerinde kodlanması için Gramian Açısıl Alan (GAF) yöntemini kullanmış, ardından bu yöntemle elde ettikleri görüntülerde önemli sekiz formasyonun tespiti için CNN yönteminden faydalanmışlardır. Bu iki adımlı yöntemin LSTM yönteminden daha yüksek bir performans sergilediğini ifade etmişlerdir. Daha sonraki çalışmalarında (Chen ve Tsai, 2022) önerdikleri yaklaşımın daha genel olarak nesne tanımda kullanılması için, bu yaklaşıma dayanarak YOLO 1 modeline bazı değişiklikler önermişlerdir. Chen ve diğerleri (2020) benzer bir yaklaşımla öncelikle zaman serilerinin kodlanması için Gramian Açısıl Toplama Alanı (GASF) yöntemini kullanmış, ardından sekiz popüler mum formasyonunun tespiti için CNN yöntemini kullanmışlardır. Önerilen GASF-CNN yöntemi ile formasyon belirlemede başarılı sonuçlar elde edilmekle birlikte, modelin insan gözüyle görülen özellikleri öğrenip öğrenmediğini anlamak amacıyla Yerel Arama Saldırısı Çekişmeli Modelini kullanmışlardır. Sonuçta önerilen modelin mum formasyonlarını yatırımcıya benzer şekilde algıladığını tespit etmişlerdir. Daha sonra yaptıkları çalışmada (Chen ve diğerleri, 2022), önerilen yönteminin yorumlanabilirliğinin, CNN’in avantajlarını kaybetmeksizin doğruluktan biraz feragat edilerek oldukça geliştirilebileceğini, dijital para birimine ait mum formasyonlarını kullanarak göstermişlerdir.

Derin öğrenme yöntemleri ile mum grafikleri analiz edilirken bir diğer yaklaşım hisse senedi fiyatlarını tahmin etmektir (Kim ve Kim, 2019; Liu ve diğerleri, 2022). Kim ve Kim (2019), yaptıkları çalışmada, SPDR S&P 500 ETF verisine ait hisse fiyatlarını tahmin etmek için zaman serileri ve mum grafiklerini birlikte kullanmışlardır. Bu amaçla LSTM ile CNN yöntemlerinin birleşimi ile hibrit bir model önerilmiş ve önerilen hibrit

yaklaşımın tekli yöntemlere göre daha başarılı tahminler verdiği sonucuna ulaşmışlardır. Liu ve diğerleri (2022) ise benzer bir yaklaşımla Evrişimli Otomatik Kodlayıcı (CAE) ile LSTM yöntemlerini kullanarak hibrit bir yaklaşım önermişler ve önerilen yaklaşımın yalnızca LSTM kullanılan modele göre daha başarılı olduğu sonucuna ulaşmışlardır.

Hisse senedi için ileriye yönelik öngörü yapılırken, doğrudan fiyatı tahmin etmek yerine fiyatın hareketi de tahmin edilebilir. Çünkü doğrudan fiyatın kendisi kadar, fiyatın düşüp yükselme eğilimi de yatırımcı için karar almada etkili bir bilgidir. Ayrıca yöntem açısından bakıldığında, fiyatın kendisi yerine hareket eğilimini öngörmek, bir sınıflandırma problemi olarak ele alınabilir. Literatürde mum grafikleri kullanılarak hisse senedinin fiyat hareketliliğini tahmin etmek için derin öğrenme yöntemlerinin kullanıldığı pek çok çalışma mevcuttur (Quan, 2013; Tsai ve Quan 2014; Guo ve diğerleri, 2018; Kusuma ve diğerleri, 2019; Jearanaitanakij ve Passaya, 2019; Andriyanto ve diğerleri, 2020; Hung ve diğerleri, 2020; Hung ve Chen, 2021). Quan (2013) ile Tsai ve Quan (2014) yaptıkları çalışmalarda, mum grafiklerinden yedi farklı dalgacık tabanlı özellik çıkarımı yapmak için görüntü alma yaklaşımı kullanmışlardır. Ardından benzer geçmiş mum grafikleri üzerinden gelecek fiyat hareketliliklerini tahmin etmişlerdir. Çalışmalarda Dow Jones Endüstriyel Ortalama Endeksi için farklı öngörü uzunluklarında tahminlere yer verilmiştir ve alternatif klasik yöntemlerle makine öğrenmesi yöntemlerine göre daha başarılı sonuçlar elde edilmiştir. Guo ve diğerleri (2018), Tayvan borsasında fiyat hareketliliğini öngörmek için CNN otomatik kodlayıcı yöntemi kullanırken; Kusuma ve diğerleri (2019) ise Tayvan ve Endonezya borsalarındaki fiyat hareketlilikleri için CNN ile çeşitli sinir ağı modellerini karşılaştırmış ve CNN modelinin daha yüksek performans sergilediğini ortaya koymuşlardır. Andriyanto ve diğerleri (2020), Endonezya maden sektör indeksinde mum grafikleri görüntülerinden fiyat hareketliliğini tahmin etmek amacıyla CNN modeli kullanmışlardır. Hung ve diğerleri (2020), mum grafiklerinden alt grafikleri ayırıştırma, alt grafiklerin en iyi temsilini elde etmek için CNN otomatik kodlayıcı kullanma ve fiyat hareketliliğini öngörme için CNN modelini kullanmayı içeren üç aşamalı bir yöntem önermişlerdir. Daha sonraki çalışmalarında Hung ve Chen (2021) ise bu üç aşamalı yöntemde değişikliğe giderek üçüncü aşamada CNN yerine RNN kullanmışlardır. Bu yöntemi kullanarak Asya

borsalarından Tayvan ve Japon borsaları için fiyat hareketliliği öngörüsünde bulunmuş ve alternatif yöntemlere kıyasla daha başarılı sonuçlar elde etmişlerdir.

Derin öğrenme yöntemleri sayısal verilerin yanı sıra, görüntü ve metin gibi farklı veri türlerini de analiz edebilme avantajına sahiptir. Bu avantajdan yola çıkılarak, fiyat hareketliliğini tahmin ederken alternatif veri türlerinden yararlanan çalışmalar mevcuttur (Ho ve Huang, 2021; Mousa ve diğerleri, 2021). Bu amaçla mum grafiklerine ait görüntü verileri ile birlikte kullanılan bir diğer veri türü, özellikle sosyal medyadan elde edilen metin verileridir. Bu yaklaşımın kullanıldığı çalışmalardan birinde Ho ve Huang (2021), mum grafikleri ile birlikte Twitter verilerinden duygu analizi kullanılarak elde edilen metin verilerini bir arada analiz etmişlerdir. Bu verilerle 4, 6, 8 ve 10 günlük periyotlar için fiyat hareketliliklerini tahmin etmek üzere bir ve iki boyutlu CNN modellerini kullanmışlardır. Önerilen model yüksek talep gören beş hisse senedi (Apple, Tesla, IBM, Amazon ve Google) için uygulanmış ve yalnızca metin veya yalnızca görüntü kullanan tekli modellere oranla daha başarılı sonuçlar elde edildiği vurgulanmıştır. Mousa ve diğerleri (2021) ise benzer şekilde hem mum görüntüleri hem de finansal metinleri birlikte kullanarak fiyat hareketliliğini öngörmek amacıyla Kapsül Ağları kullanmayı önermişlerdir. Kapsül ağlarının görüntü verilerindeki uzamsal ilişkileri ve metinlerdeki anlamsal ilişkileri koruması özelliği dikkate alınmış ve euro/dolar hisse senedi davranışı için tahmin yapılmıştır.

Hisse senedi için fiyat hareketliliğini tahmin etmedeki temel amaç yatırımcılar için karar vermeyi kolaylaştıran destekleyici bilgi elde edebilmektir. Bu nedenle, fiyat hareketliliği çalışmaları doğrudan bir sonraki aşamaya taşınarak alım - satım veya bekletme kararı vermeye evrilebilir. Bir başka deyişle, hisse senedinin fiyatını öngörmek yerine, hisse senedi için doğrudan al - sat - bekle kararı verilebilir. Derin öğrenme açısından ele alındığında bu bir sınıflandırma problemi haline gelmektedir. Derin öğrenme yöntemleri kullanılarak yatırım kararı verebilmek için modellerin önerildiği çeşitli çalışmalar bulunmaktadır (Sezer ve Özbayoğlu, 2018, 2019; Biroğul ve diğerleri, 2020; Santur, 2022). Sezer ve Özbayoğlu (2018, 2019) yaptıkları çalışmalarda öncelikle hisse senetlerine ait 15 farklı teknik göstergiyi dikkate alarak iki boyutlu görüntüler elde etmişler ve bu görüntüleri “al”, “sat”, “bekle” olarak etiketlemişlerdir. Bu etiketli veriler

kullanılarak CNN yöntemi yardımıyla al -sat- bekle kararı vermeye yönelik bir model önermişlerdir. Daha sonraki çalışmalarında ise teknik göstergeler yerine hisse senetlerine ait çubuk grafiklerinin iki boyutlu görüntüleri için benzer bir karar destek modeli tasarlamışlardır. Her iki çalışmalarında da önerilen modellerin özellikle uzun öngörü periyotları için geleneksel al & sat stratejilerine oranla daha başarılı sonuçlar verdiğini ortaya koymuşlardır. Biroğul ve diğerleri (2020), yaptıkları çalışmada al - sat karar destek modeli oluşturmak için mum grafiklerini girdi olarak almış ve nesne tanıma algoritması olan YOLO algoritmasıyla CNN yöntemini birlikte kullandıkları bir model önermişlerdir. Model 2000 - 2018 yılları arasındaki Borsa İstanbul (BİST) verilerine uygulanmış ve % 85’lik bir başarı elde edilmiştir. Santur (2022) ise al - sat karar modeli oluşturmak için mum grafiklerini etiketledikten sonra XGBoost algoritmasından faydalanmıştır. Önerilen model BİST30 da dahil olmak üzere çeşitli dünya piyasaları için uygulanmıştır.

Yatırımcılar için karar destek modelleri tek bir hisse senedi için al - sat kararı olarak tasarlanabileceği gibi, daha geniş kapsamlı olarak karlılığı maksimize etmeye yönelik portföy yönetim modelleri olarak da tasarlanabilir. Literatürde derin öğrenme yöntemlerinin bu amaçla kullanıldığı çalışmalara rastlanmaktadır (Hu ve diğerleri, 2018; Pan ve diğerleri, 2020; Li ve diğerleri, 2022; Liu ve diğerleri, 2023). Hu ve diğerleri (2018), bir yatırım karar strateji modeli oluşturmak için Evrimsel Otomatik Kodlayıcıdan (CAE) faydalanmışlardır. Mum grafiklerinin girdi olarak alındığı modelde, CAE yardımıyla özellik çıkarımı yapılmış ve benzer özelliklerin kümelenmesi için modülerlik optimizasyonu kullanılmıştır. Ardından portföy inşası için her kümeden Sharpe oranına göre en yüksek performansa sahip hisseler seçilmiştir. Sonuçta toplam getiri açısından FTSE 100 indeksinden ve bilindik fonlardan daha başarılı bir performans elde edilmiştir. Benzer bir yaklaşımın uygulandığı Pan ve diğerleri (2020) çalışmasında, otomatik kodlayıcının ardından kümeleme için K-ortalamlar kullanılmış ve portföy seçimi için yine Sharpe oranından faydalanılmıştır. Çin borsası için uygulanan model sonucunda çeşitli fonlardan daha başarılı sonuçlar elde edilmiştir. Li ve diğerleri (2022) ile Liu ve diğerleri (2023) çalışmalarında ise yatırım stratejisi oluşturmak için derin takviyeli öğrenme modelleri kullanılmıştır. Hisse verisi, mum grafikleri ve teknik göstergeleri içeren çoklu veri türleri için CNN ve LSTM gibi yöntemler takviyeli öğrenme (reinforcement learning) yöntemleri ile bir araya getirilerek derin takviyeli

öğrenme modelleri oluşturulmuştur. Önerilen modeller yardımıyla Çin ve S&P 500 borsaları için Sharpe oranına göre klasik fonlardan daha yüksek getiriye sahip portföyler elde edilmiştir.

4.3 Veri Seti ve Metodoloji

Bu çalışmada mum formasyonların görüntüleri kullanılarak hisse senetlerindeki fiyat ve fiyat hareketliliği tahmini yapılması amaçlanmıştır. Bu amaç doğrultusunda, öncelikle CNN algoritması kullanılarak verilen bir mum grafiğinin hangi formasyona ait olduğunun sınıflandırılması hedeflenmiştir. İlk olarak, CNN algoritmasının görüntüleri tespit edebilmesi için, mum grafiklerine ve bu grafiklerin formasyon bilgilerinin etiketlerine ihtiyaç duyulmaktadır. Bu sebeple, BİST30’da yer alan hisse senetlerinin günlük açılış, kapanış, en yüksek ve en düşük hisse senedi fiyat değerleri dikkate alınmıştır. Çalışmaya dahil olan hisse senetlerinin listesi Tablo 2’de verilmiştir.

Tablo 2
Hisse Senedi Kodları

AKBNK	ARCLK
ASELS	BIMAS
BRMEN	CMENT
DENGE	EKGYO
EREGL	FROTO
GARAN	GUBRF
HALKB	HEKTS
ISBIR	ISCTR
ISYAT	KCHOL
KENT	KLNMA
KOZAA	KOZAL
KRDMD	PETKM
PGSUS	QNBFB
QNBFL	SAHOL
SASA	SISE
SNKRN	TAVHL
TBORG	TCELL
THYAO	TKFEN
TOASO	TTKOM
TUPRS	YKBNK
VESTL	ZEDUR

Tablo 2’de yer alan hisse senetlerinin açılış, kapanış, en düşük ve en yüksek fiyat değerleri *Yahoo Finance* veri tabanından elde edilmiş olup, verilerin elde edilme sürecinde, Python’da yer alan “yfinance” kütüphanesinden yararlanılmıştır. Söz konusu kütüphane yardımıyla, her hisse senedinin borsada işlem gördüğü kod kullanılarak günlük verileri elde edilmiştir. CNN modelinde yüksek öğrenme başarısının sağlanabilmesi için veri sayısının yüksek olması gerektiğinden, çalışmada periyod olarak alınabilecek maksimum süre alınmıştır. Listedeki ilk hisse senedinden başlayarak sırasıyla diğer tüm hisse senetlerine ait fiyatların çekilmesi sağlanmıştır. Veri seti, 22.07.2005 -22.07.2023 tarihleri arasında her hisse senedinin BİST30’a dahil olduğu tarihten itibaren alınarak oluşturulmuştur.

CNN algoritmasının çalıştırılabilmesi için, elde edilen fiyat verilerinin mum grafiklerine dönüştürülmesi ve etiketlenmesi gerekmektedir. Mum formasyonlarına ait etiketli hazır bir veri setinin olmaması bir sorun teşkil etmektedir. Bu formasyonların araştırmacı tarafından manuel olarak etiketlenmesi hem zaman hem de işlem maliyetine neden olmakta, aynı zamanda hatalı etiketleme yapılması ihtimalini ortaya çıkarmaktadır. Bu nedenle, verilerin etiketlenme işleminin otomatik olarak gerçekleştirilmesi için Python kütüphanelerinden faydalanılmıştır. Öncelikle “ploty” kütüphanesi ile mum grafikleri çizilmiş sonrasında TA-Lib kütüphanesinin fonksiyonları kullanılarak etiketleme işlemi gerçekleştirilmiştir.

Bu çalışmada, çok sayıda mum formasyonu olmakla birlikte, fiyat hareketliliğinde dönüşümü gösteren ve literatürde sıklıkla yer alan üçlü formasyonlara odaklanılmıştır. Bu sebeple, fiyat verilerinden hareketle üçlü mum grafiklerinin çizdirilerek formasyonların görüntülerinden oluşan veri seti elde edilmiştir. Bu işlem yapılırken birinci günden başlanarak ve birer gün kaydırmalı olacak şekilde üçlü mum grafikleri döngüsel olarak çizdirilmiştir. Grafiklerin genişlik ve yükseklikleri 224x224 piksel olarak ayarlanmıştır. Bu değerin azaltılması grafiğin okunaklılığını azaltmakta, dolayısıyla makinenin öğrenmesini zorlaştırmakta, yükseltilmesi ise öğrenme sürecinin uzamasına neden olmaktadır. Bu sebeple söz konusu değer denemeler sonucunda ideal olarak 224 piksel olarak belirlenmiştir. Grafiklerin çizimi için TA-Lib kütüphanesinde yer alan *Candlestick* fonksiyonu kullanılmıştır.

Söz konusu grafikler ile birlikte grafiğin hangi formasyona ait olduğunun da tespit edilmesi gerekmektedir. Bu etiketleme işlemi için ise, TA-Lib kütüphanesindeki *Pattern Recognition* fonksiyonu kullanılmıştır. Bu fonksiyon sayısal verileri kullanarak, ilgili tarihte herhangi bir formasyonun olup olmadığını ve eğer varsa, ilgili formasyonun ayı ve boğa piyasalarından hangisine ait olduğunu belirlemektedir. TA-Lib kütüphanesi yardımıyla tekli, birli, ikili, üçlü vb. olmak üzere 61 adet farklı formasyon tespit edilebilmektedir. Bu çalışmada fiyatların hareketliliğinde belirleyici olan üçlü formasyonlara odaklanıldığından, akşam yıldızı, sabah yıldızı, üç içerde yukarı/aşağı, üç dıştan yukarı/aşağı, üç yıldız yukarı/aşağı formasyonları kullanılmıştır. Bu formasyonların de boğa ve ayı durumları da dikkate alındığında 8 farklı görüntü analize dahil edilmiştir. TA-Lib kütüphanesinden elde edilen formasyon bilgisi kullanılarak analize dahil edilen üçlü mum grafikleri etiketlenmiştir. Tespit edilen etiketlemeler ilgili görüntünün ismi olarak kullanılmıştır. Bu şekilde 8 mum formasyonuna ait etiketli görüntüler elde edilmiştir. Etiketli verilerin yer aldığı klasörler Şekil 63'te verilmiştir.



Şekil 63: Etiketli Mum Formasyonları Klasörleri

BİST30'da yer alan hisse senetlerinin formasyon görüntülerini sınıflandırmak için CNN modeli uygulanmış ve ilk olarak modelin öğrenme aşaması için veri seti eğitim, geçerlilik ve test olmak üzere üç gruba ayrılmıştır. Bu ayrımında, her hisse senedinin BİST30 endeksine dahil olduğu tarih farklı olduğundan, standart bir aralık belirlemek amacıyla her hisse senedinin son 250 günü test verisi olarak ayrılmıştır. Geriye kalan günlere ait veriler birleştirilerek model kurulması için kullanılmıştır. Daha sonra birleştirilmiş veri seti eğitim (train) ve geçerlilik (validation) için sırasıyla %70'e %30 oranında rassal olarak ayrılmıştır.

CNN mimarisi oluşturulurken konvolüsyon, havuzlama, düzleştirme, yoğunluk ve seyreltme katmanları kullanılmıştır. Bu modelde seyreltme katmanının kullanılmasının amacı aşırı öğrenmeden kaçınmaktır. Modelde aktivasyon fonksiyonları olarak ReLU ve softmax fonksiyonları kullanılmıştır. Modelde yer alan girdiler pozitif olduğundan, fonksiyonunun hızlı eğitme ve düşük hesaplama maliyeti avantajlarından faydalanmak amacıyla ReLU fonksiyonu tercih edilmiştir. Softmax fonksiyonu ise, çoklu sınıflandırma uygulamalarında başarılı olduğu için çıktı katmanında tercih edilmiştir. Modeldeki kayıp fonksiyonu olarak softmax çapraz entropi (categorical crossentropy) ve optimizasyon algoritması olarak ADAM algoritması kullanılmıştır. Model başarı kriteri olarak doğruluk değeri ve kayıp fonksiyonu kullanılmıştır. Doğruluk değeri,

$$\text{Doğruluk} = \frac{\text{Gerçek Pozitif} + \text{Gerçek Negatif}}{\text{Gerçek Pozitif} + \text{Gerçek Negatif} + \text{Sahte Pozitif} + \text{Sahte Negatif}}$$

olarak hesaplanmaktadır. CNN modelinde optimal katman sayıları ve hiperparametre değerlerinin elde edilmesi amacıyla çok sayıda model denenmiş ve elde edilen en uygun model tercih edilmiştir. Belirlenen modele ait optimal hiperparametre değerleri bulgular kısmında yer almaktadır.

CNN modelinde formasyonların 8 sınıftan birine sınıflandırılması işlemi yapılmaktadır. Burada formasyonun her bir sınıfa ait olup olmaması ikili (binary) olarak analize dahil edilebilir. Ancak formasyonlar nadiren ortaya çıktığından formasyonların yokluk bilgisi analize eklendiğinde, bu durum derin öğrenme algoritmalarının sadece baskın olan yokluk bilgisini öğrenmesine sebep olmakta ve bu da hatalı tahminlerin yapılmasına yol açmaktadır. Bu sorunun aşılması için istatistiksel temele dayanan alternatif bir yaklaşım tercih edilmiştir. Bu yaklaşımda formasyonun her bir sınıfa dahil olup olmaması bilgisi yerine sınıflara atanma olasılıkları alınmıştır. Böylece sıfır ağırlıklı matris sorunu aşılmıştır. Elde edilen olasılık değerleri kapanış fiyatlarının tahmininde kullanılmıştır.

Çalışmanın ikinci kısmında, hisse senetlerinin gelecek dönem kapanış fiyatlarının ve bu fiyatların hareket yönlerinin tahmin edilmesi aşamasına geçilmiştir. Burada formasyon bilgisinin fiyatları ve fiyat hareketliliğini tahmin etmede etkili olup olmadığı incelenmiştir. Bu amaçla, CNN algoritmasından elde edilen formasyonlara ait

sınıflandırma olasılıkları ile bir veri seti oluşturulmuştur. Ayrıca sadece formasyon bilgisi kullanmak yerine, karşılaştırmak amacıyla klasik literatürde yer alan fiyatların açılış, kapanış, günün en düşük/en yüksek değeri ve işlem hacmi bilgilerini içeren ikinci veri seti oluşturulmuştur. Daha sonra hem formasyon hem de fiyat bilgisinin birlikte kullanılmasının fiyat tahmininde daha etkili olup olmadığını incelemek amacıyla, bu iki veri seti birleştirilerek üçüncü veri seti oluşturulmuştur. Sadece olasılıkların yer aldığı veri seti *olasılık*, fiyatların yer aldığı veri seti *fiyat*, her ikisinin birleşiminden oluşan veri seti ise *olasılık_fiyat* veri seti olarak isimlendirilmiş ve bulgular kısmında bu şekilde kullanılmıştır. Her üç veri seti için kapanış fiyatı ve fiyat hareketliliğinin yönü tahmin edilmiş ve sonuçlar karşılaştırmalı olarak yorumlanmıştır.

Hisse senetlerine ait kapanış fiyatı ve fiyat hareketliliğinin yönünü tahmin etmek amacıyla yinelemeli sinir ağlarından LSTM ve GRU algoritmaları kullanılmıştır. Bu aşamada CNN'den farklı olarak her bir hisse senedi ayrı ayrı eğitim sürecinden geçmektedir. CNN modelinde tüm hisse senetlerindeki formasyonlar ortak bir veri havuzu oluşturularak eğitim gerçekleştirilmiş, ardından test aşaması ayrı ayrı hisse senetleri için yapılmıştır. LSTM ve GRU uygulamalarında ise CNN algoritmasında doğru sınıflandırma oranı en yüksek olan hisse senetlerinden 3 tanesi seçilerek 3 ayrı eğitim ve tahmin gerçekleştirilmiştir. Modelin farklı durumlara karşı duyarlılığını incelemek amacıyla seçilen üç hisse senedinden birisi çok sayıda formasyona sahipken, diğeri daha az sayıda formasyona sahiptir. Üçüncü hisse senedi ise aldığı değerler bakımından daha geniş bir aralığa sahiptir. Ancak üç hisse senedinde ortak nokta, formasyonların doğru sınıflandırma oranı %100 olan hisse senetleri arasından seçilmiş olmalarıdır. Burada her bir hisse senedi için veri setinin %80'i eğitim verisi, %20'si geçerlilik verisi olarak alınmıştır. Test verisi olarak ise bu sürece dahil olmayan sonraki 10 gün alınmıştır.

Hem veri setleri hem de algoritmaların performansını karşılaştırmak amacıyla algoritmalarda standart bir model yapısı oluşturmak hedeflenmiş ve ortak hiperparametreler kullanılmıştır. Bu ortak hiperparametrelerden geçmiş pencere (*hist_window*) boyutu 90, tahmin ufku (*horizon*) ise 10 olarak alınmıştır. Böylece her adımda geçmiş 90 değerden hareketle gelecek 10 değer tahmin edilerek model eğitilmiştir. Formasyonların etkilerinin kısa dönemde görülmesi sebebiyle gelecek dönem

tahmin uzunluğu yüksek tutulmamıştır. Ayrıca hiperparametrelerden yığın boyutu (batch size) 64, tampon boyutu (buffer size) ise 150 olarak alınmıştır. Algoritmelerde LSTM ve GRU katmanlarına ek olarak seyreltme ve yoğunluk katmanları kullanılmıştır. Aktivasyon fonksiyonu olarak hiperbolik tanjant fonksiyonu kullanılırken, optimizasyon algoritması olarak ADAMW algoritması, kayıp fonksiyonu olarak ise MSE kullanılmıştır. Modellerin performans karşılaştırılmasında ise RMSE ve MAE metrikleri kullanılmıştır. Bu metrikler aşağıdaki gibi hesaplanmaktadır:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

Her üç veri seti için iki algoritma ile kapanış fiyatları tahmin edilmiştir. Ayrıca fiyatların artış azalış yönleri için tahmin edilen fiyatlardan hareketle günlük olarak hareket yönü incelenmiş ve gerçek hareket yönleri ile karşılaştırılmıştır. Her bir model ve veri seti için hareket yönünün doğru tahmin edilme oranları hesaplanmıştır. Böylece modellerin fiyatların hareket yönünü yakalama performansları da incelenmiştir. Algoritmaların detayları ve karşılaştırma sonuçları bulgular kısmında sunulmuştur.

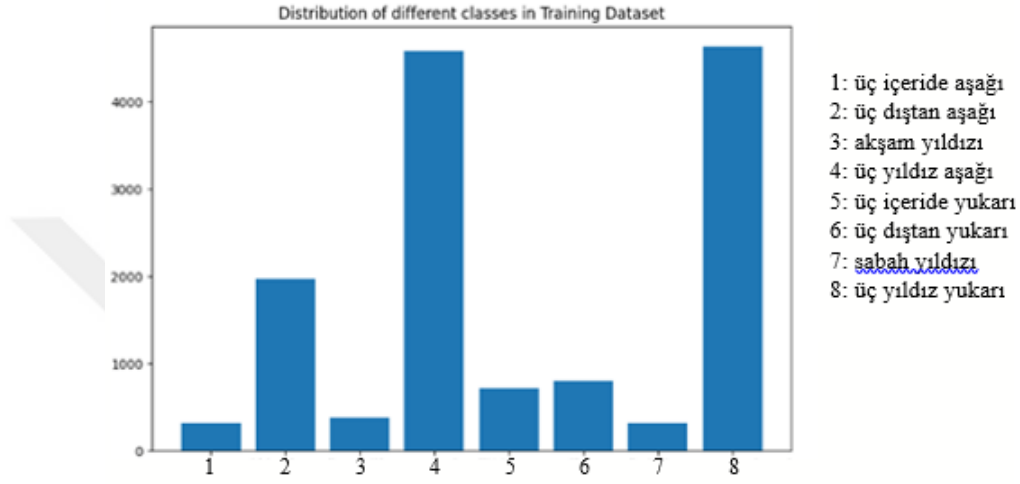
4.4 Bulgular

Bu bölümde bulgular iki aşamada sunulmuştur. İlk kısımda analizdeki mum formasyonlarının sınıflandırılmasında kullanılan CNN algoritmasının bulgularına yer verilmiştir. İkinci kısımda ise CNN algoritmasının sınıflandırma olasılıklarından hareketle oluşturulan veri setlerinin LSTM ve GRU bulguları sunulmuştur.

4.4.1 CNN Bulguları

Mum formasyonlarına ait görüntülerin sınıflandırılması için CNN algoritması kullanılmıştır. Metodolojide anlatıldığı şekilde etiketlenen formasyon görüntüleri alınarak CNN algoritması ile sınıflandırma işlemi yapılmıştır. Bu işlem için 20253 adet

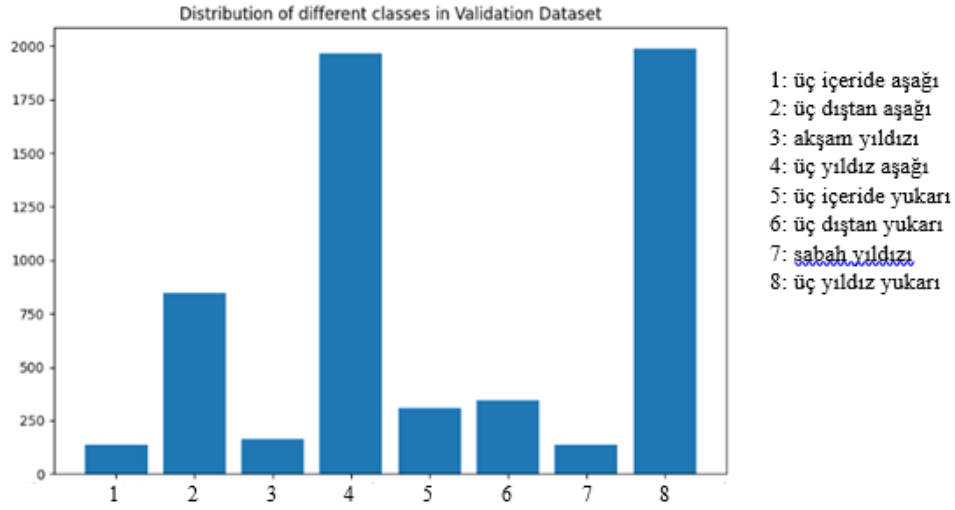
resim elde edilmiştir. CNN algoritmasının girdisi olarak kullanılan etiketli veriler eğitim, geçerlilik ve test verisi olarak ayrılmıştır. Test verisi hariç geriye kalan veri setinde 19601 adet etkilenmiş resim elde edilmiş, eğitim veri seti olarak 13717 adet etiketli resim kullanılmıştır. Test verisi olarak 652 adet resim ayrılmıştır. Sekiz mum formasyonunun bu veri setindeki dağılımı Şekil 64’te verilmiştir.



Şekil 64: Eğitim Veri Setindeki Formasyonların Dağılımı

Şekil 64 incelendiğinde, üç içeride aşağı formasyonundan 319, üç dıştan aşağı formasyonundan 1966, akşam yıldızı formasyonundan 378, üç yıldız aşağı formasyonundan 4582, üç içeride yukarı formasyonundan 721, üç dıştan yukarı formasyonundan 801, sabah yıldızı formasyonundan 319 ve üç yıldız yukarı formasyonundan 4631 resim eğitim verisi olarak kullanıldığı görülmektedir.

Benzer şekilde geçerlilik veri setinde ayırışma sonunda 5884 adet etkilenmiş resim elde edilmiştir. Geçerlilik için kullanılan formasyonların dağılımı Şekil 65’te verilmiştir.



Şekil 65: Geçerlilik Veri Setindeki Formasyonların Dağılımı

Şekil 65 incelendiğinde, üç içeride aşağı formasyonundan 137, üç dıştan aşağı formasyonundan 844, akşam yıldızı formasyonundan 163, üç yıldız aşağı formasyonundan 1964, üç içeride yukarı formasyonundan 309, üç dıştan yukarı formasyonundan 344, sabah yıldızı formasyonundan 137 ve üç yıldız yukarı formasyonundan 1986 resim eğitim verisi olarak kullanıldığı görülmektedir.

Eğitim ve geçerlilik veri setleri kullanılarak CNN modelinin eğitilmesi sağlanmıştır. Kullanılan CNN mimarisi aşağıdaki gibidir:

```
model = Sequential([
    Conv2D(16, (3, 3), activation='relu', input_shape=(img_height, img_width, 3)),
    MaxPooling2D(2, 2),
    Conv2D(32, (3, 3), activation='relu'),
    Conv2D(64, (3, 3), activation='relu'),
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D(2, 2),
    Conv2D(128, (3, 3), activation='relu'),
    Conv2D(128, (3, 3), activation='relu'),
    MaxPooling2D(2, 2),
    Conv2D(256, (3, 3), activation='relu'),
    Conv2D(256, (3, 3), activation='relu'),
    MaxPooling2D(2, 2),
    Flatten(),
    Dense(512, activation='relu'),
    Dropout(0.1),
    Dense(512, activation='relu'),
    Dense(8, activation='softmax')
])
```

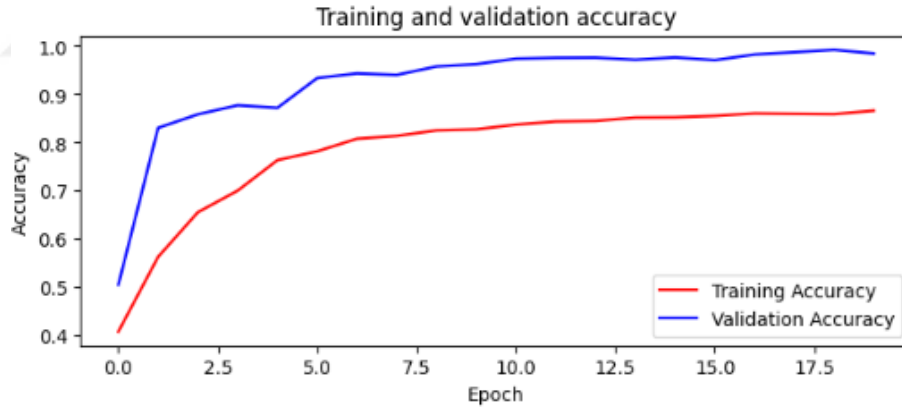
CNN modelinde öncelikle bir adet konvolüsyon katmanı oluşturulmuştur. Bu katman, 16 adet 3'er adımli giriş katmanıdır. Bununla birlikte havuzlama katmanı ile boyut azaltılmıştır. Aktivasyon fonksiyonu olarak ReLU kullanılmıştır. Benzer şekilde, aktivasyon fonksiyonu ReLU olan üç adet konvolüsyon katmanı daha eklenmiştir. Bu katmanların sonunda tekrar bir havuzlama katmanı daha eklenmiş ve boyut azaltılmıştır. Ardından yine aktivasyon fonksiyonu ReLU olan iki konvolüsyon katmanı ve havuzlama katmanları eklenmiştir. En son konvolüsyon işlemleri için iki ReLU aktivasyon fonksiyonlu konvolüsyon katmanı eklenmiştir. Elde edilen sonuçlarla düzleştirme katmanı eklenmiştir. Düzleştirme katmanından sonrada derin öğrenme için ikisi ReLU, sonuncusu softmax aktivasyon fonksiyonuna sahip üç yoğunluk (dense) katmanı eklenmiştir. Bu katmanlarda sırasıyla 512, 512 ve 8 nöron kullanılmıştır. Araya aşırı öğrenmeden kaçınmak için seyreltme (dropout) katmanı da eklenmiştir. CNN modelinin özeti ve katmanlarda kullanılan parametre sayıları Tablo 3'te verilmiştir.

Tablo 3
CNN Modeli Çıktı Yapısı

CNN Modeli		
Katman(tür)	Çıktı Yapısı	Parametre
Conv	(None, 222, 222, 16)	448
max_pooling2d	(None, 111, 111, 16)	0
conv2d	(None, 109, 109, 32)	4640
conv2d	(None, 107, 107, 64)	18496
conv2d	(None, 105, 105, 64)	36928
max_pooling2d	(None, 52, 52, 64)	0
conv2d	(None, 50, 50, 128)	73856
conv2d	(None, 48, 48, 128)	147584
max_pooling2d	(None, 24, 24, 128)	0
conv2d	(None, 22, 22, 256)	295168
conv2d	(None, 20, 20, 256)	590080
max_pooling2d	(None, 10, 10, 256)	0
Flatten	(None, 25600)	0
Dense	(None, 512)	13107712
Dropout	(None, 512)	0
dense	(None, 512)	262656
dense	(None, 8)	4104
Toplam Parametre: 14.541.672 (55.47 MB)		
Eğitilebilir Parametre: 14.541.672 (55.47 MB)		
Eğitilemeyen Parametre: 0 (0.00 Byte)		

Tablo 3'e göre, tasarlanan CNN model çıktısında her bir aşamada kaç parametre kullanıldığı yer almaktadır. Oluşturulan model, toplam 14.541.672 adet parametre içermektedir. Bu parametrelerin tamamı önceden belirlenmemiş olup, eğitilebilir durumdadır. Başka bir ifade ile, parametrelerin tümünün eğitim aşamasında değerleri güncellenebilmektedir. Oluşturulan bu modelin optimizasyonu için ADAM algoritması ve kayıp fonksiyonu için de Categorical (softmax) Crossentropy kullanılmıştır. Başarı metriği olarak doğruluk (accuracy) değeri baz alınmıştır.

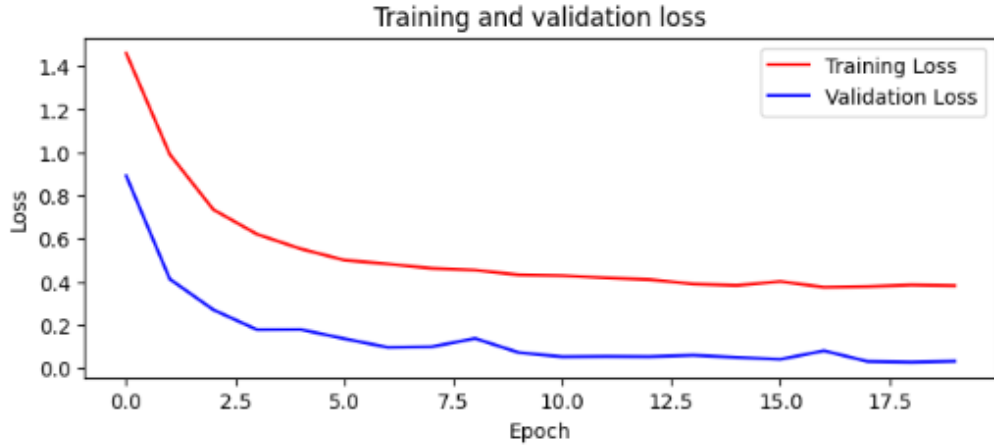
Model inşa edildikten sonra eğitim süreci başlamıştır. Başka bir ifade ile model parametrelerinin hesaplanması yapılmıştır. Döngü (epoch) değeri 20 olarak belirtilmiştir, yığın boyutu (batch size) değeri ise 64 olarak belirlendiğinden, yani her adımda 64'er resimle eğitim gerçekleşeceğinden her bir döngü toplam eğitim verisi sayısının 64'e bölümünden hesaplanan ($13717/64=214,33$) 215 adım içermektedir. 20 döngü sonunda, eğitim için doğruluk oranı %86,62, geçerlilik için doğruluk oranı %98,45 olarak hesaplanmıştır. Her bir döngüdeki doğruluk oranının değişimi Şekil 66'da verilmiştir.



Şekil 66: Eğitim ve Geçerlilik İçin Doğruluk Oranı

Şekil 66 incelendiğinde, eğitim veri setinde %40,87 ile başlayan modelin öğrenme başarısı, 20 döngü tamamlandığında %86,62'ye ulaşmış ve model başarılı bir şekilde eğitilmiştir. Benzer durum geçerlilik verileri için de görülebilmektedir.

Modelin diğ er bir parametresi olan kayıp fonksiyonu değeri ise son d ong de eđitim verisi i in 0.3737 iken ge erlilik verisi i in 0.0411 olarak hesaplanmıřtır. Her bir d ong deki kayıp fonksiyonu değeri řekil 67’de verilmiřtir.



řekil 67: Eđitim ve Ge erlilik i in Kayıp Fonksiyonu

řekil 67 incelendiđinde, eđitim verisi i in 1.4501 ile bařlayan kayıp fonksiyonu değeri 20 d ong n n sonunda 0.3737’e kadar d řm řt r. Ge erlilik verileri i in de benzer durum ge erlidir.

Burada elde edilen model uygun model olup, test verileri i in kullanılmıřtır. Daha  nce belirtildiđi gibi her bir hisse senedinin son 250 g n ne ait verilerden elde edilen mum formasyonu g r nt leri test verisi olarak alınmıř ve model uygulanmıřtır. Oluřturulan CNN modeli yardımıyla test verilerinde formasyonların sınıflandırılma iřlemi ger ekleřtirilmiřtir. S z konusu bu iřlem Tablo 2’de yer alan t m hisse senetlerinin son 250 g nl k verisindeki TA-Lib k t phanesi ile etiketlenebilmiř g nler i in yapılmıřtır.  rnek olarak AKBNK hisse senedinin detaylı sonu ları Tablo 4’de yer almaktadır.

Tablo 4
AKBNK Verilerinin Sonuçları

AKBNK	Model Sonucu	Gerçek Değer
1	Üç İçeride Yukarı	Üç İçeride Yukarı
2	Üç İçeride Aşağı	Üç İçeride Aşağı
3	Üç İçeride Yukarı	Üç İçeride Yukarı
4	Üç İçeride Yukarı	Üç İçeride Yukarı
5	Üç Dıştan Yukarı	Üç Dıştan Yukarı
6	Üç Dıştan Aşağı	Üç Dıştan Aşağı
7	Üç Dıştan Aşağı	Üç Dıştan Aşağı
8	Üç Dıştan Aşağı	Üç Dıştan Aşağı
9	Üç Dıştan Aşağı	Üç Dıştan Aşağı
10	Üç Dıştan Yukarı	Üç Dıştan Yukarı
11	Üç Dıştan Yukarı	Üç Dıştan Yukarı
12	Üç Dıştan Aşağı	Üç Dıştan Aşağı
13	Üç Dıştan Aşağı	Üç Dıştan Aşağı
14	Üç Dıştan Aşağı	Üç Dıştan Aşağı
15	Üç Dıştan Aşağı	Üç Dıştan Aşağı
16	Üç Dıştan Aşağı	Üç Dıştan Aşağı
17	Akşam Yıldızı	Akşam Yıldızı
18	Sabah Yıldızı	Sabah Yıldızı
19	Sabah Yıldızı	Sabah Yıldızı
20	Üç Yıldız Yukarı	Üç Yıldız Aşağı

Tablo 4 incelendiğinde son 250 günde 20 adet formasyon tespit edilmiştir. Söz konusu bu formasyonların ortaya çıkış sırası ve model sonuçları Tablo 4’te verilmiştir. Model sonuçları ile gerçek değerler kıyaslandığında başarılı sınıflandırma oranının %95 olduğu görülmüştür. Model, son satırda yer alan gerçekte üç yıldız aşağı olan formasyonu, üç yıldız yukarı formasyonu olarak sınıflandırmış ve tek formasyonda hata yapmıştır. Benzer bulgular tüm hisse senetleri için elde edilmiş ve Tablo 5’de özetlenmiştir. Bulguların detayları EK-1’de yer almaktadır.

Tablo 5
Hisse Senetlerinin Test Verisi Sonuçları

		Akşam Yıldızı	Sabah Yıldızı	Üç İçeride Aşağı	Üç İçeride Yukarı	Üç Dıştan Aşağı	Üç Dıştan Yukarı	Üç Yıldız Aşağı	Üç Yıldız Yukarı	Toplam	Doğru Sınıflandırma Oranı
‘AKBNK.IS’	Model	1	2	1	3	9	3	0	1	20	%95
	Gerçek	1	2	1	3	9	3	1	0		
‘ARCLK.IS’	Model	5	3	1	1	3	2	1	1	17	%100
	Gerçek	5	3	1	1	3	2	1	1		
‘ASELS.IS’	Model	1	2	1	3	7	1	0	0	15	%100
	Gerçek	1	2	1	3	7	1	0	0		
‘BIMAS.IS’	Model	1	4	0	4	4	2	0	0	15	%100
	Gerçek	1	4	0	4	4	2	0	0		
‘BRMEN.IS’	Model	0	1	0	4	4	0	0	0	9	%100
	Gerçek	0	1	0	4	4	0	0	0		
‘CMENT.IS’	Model	1	3	1	2	4	2	0	1	14	%100
	Gerçek	1	3	1	2	4	2	0	1		
‘DENGE.IS’	Model	1	1	0	0	4	1	0	0	7	%100
	Gerçek	1	1	0	0	4	1	0	0		
‘EKGYO.IS’	Model	3	0	0	2	8	2	0	0	15	%100
	Gerçek	3	0	0	2	8	2	0	0		
‘EREGL.IS’	Model	3	1	0	2	10	2	0	0	18	%100
	Gerçek	3	1	0	2	10	2	0	0		
‘FROTO.IS’	Model	2	3	1	2	4	1	0	1	14	%93
	Gerçek	2	3	1	3	4	0	0	1		
‘GARAN.IS’	Model	3	0	1	1	7	6	0	1	19	%89
	Gerçek	3	0	1	1	7	5	1	1		
‘GUBRF.IS’	Model	2	0	0	6	10	4	0	0	22	%95
	Gerçek	2	1	0	5	10	4	0	0		
‘HALKB.IS’	Model	0	0	1	1	7	4	0	0	13	%92
	Gerçek	0	0	1	2	7	3	0	0		
‘HEKTS.IS’	Model	3	0	4	1	9	2	0	0	19	%100
	Gerçek	3	0	4	1	9	2	0	0		
‘ISBIR.IS’	Model	4	0	1	2	6	2	0	0	15	%93
	Gerçek	3	0	2	2	6	2	0	0		
‘ISCTR.IS’	Model	1	0	1	6	7	3	1	0	19	%89
	Gerçek	1	0	1	7	8	2	0	0		
‘ISYAT.IS’	Model	0	2	0	2	7	5	0	0	16	%87,5
	Gerçek	0	2	0	4	7	3	0	0		
‘KCHOL.IS’	Model	2	0	1	4	11	3	0	0	21	%95
	Gerçek	2	0	1	5	11	2	0	0		
‘KENT.IS’	Model	2	1	1	1	7	3	0	0	15	%100
	Gerçek	2	1	1	1	7	3	0	0		
‘KLNMA.IS’	Model	0	0	1	1	6	2	0	1	11	%82
	Gerçek	0	1	1	0	6	2	1	0		
‘KOZAA.IS’	Model	5	1	0	3	6	1	0	0	16	%94
	Gerçek	5	2	0	2	6	1	0	0		

Tablo 5 (devam)
Hisse Senetlerinin Test Verisi Sonuçları

		Akşam Yıldızı	Sabah Yıldızı	Üç İçeride Aşağı	Üç İçeride Yukarı	Üç Dıştan Aşağı	Üç Dıştan Yukarı	Üç Yıldız Aşağı	Üç Yıldız Yukarı	Toplam	Doğru Sınıflandırma Oranı
‘KOZAL.IS’	Model	4	0	0	8	7	0	0	0	19	%95
	Gerçek	4	1	0	7	7	0	0	0		
‘KRDMD.IS’	Model	1	0	1	1	11	2	0	0	16	%100
	Gerçek	1	0	1	1	11	2	0	0		
‘PETKM.IS’	Model	1	2	2	2	5	2	0	0	14	%93
	Gerçek	1	2	2	3	5	1	0	0		
‘PGSUS.IS’	Model	1	0	2	3	2	3	0	0	11	%91
	Gerçek	1	0	2	4	2	2	0	0		
‘QNBFB.IS’	Model	4	1	1	0	1	4	0	0	11	%91
	Gerçek	4	1	1	1	1	3	0	0		
‘QNBFL.IS’	Model	1	0	2	2	4	1	0	1	11	%100
	Gerçek	1	0	2	2	4	1	0	1		
‘SAHOL.IS’	Model	0	3	4	1	11	4	0	0	23	%100
	Gerçek	0	3	4	1	11	4	0	0		
‘SASA.IS’	Model	0	0	3	3	6	3	0	0	15	%93
	Gerçek	1	0	2	3	6	3	0	0		
‘SISE.IS’	Model	3	2	2	1	6	4	0	0	18	%100
	Gerçek	3	2	2	1	6	4	0	0		
‘SNKRN.IS’	Model	0	2	1	1	3	5	0	0	12	%100
	Gerçek	0	2	1	1	3	5	0	0		
‘TAVHL.IS’	Model	2	0	0	2	12	0	0	0	16	%100
	Gerçek	2	0	0	2	12	0	0	0		
‘TBORG.IS’	Model	1	1	0	1	5	0	0	0	8	%100
	Gerçek	1	1	0	1	5	0	0	0		
‘TCELL.IS’	Model	0	2	0	3	6	0	0	0	11	%100
	Gerçek	0	2	0	3	6	0	0	0		
‘THYAO.IS’	Model	1	0	3	1	8	1	0	0	14	%100
	Gerçek	1	0	3	1	8	1	0	0		
‘TKFEN.IS’	Model	2	1	0	8	8	3	0	0	22	%95
	Gerçek	2	2	0	7	8	3	0	0		
‘TOASO.IS’	Model	1	1	0	4	5	4	0	0	15	%87
	Gerçek	1	2	0	4	5	3	0	0		
‘TTKOM.IS’	Model	0	0	0	4	8	4	0	0	16	%81
	Gerçek	0	0	0	7	8	1	0	0		
‘TUPRS.IS’	Model	6	0	1	2	10	2	0	0	21	%100
	Gerçek	6	0	1	2	10	2	0	0		
‘VESTL.IS’	Model	0	1	1	4	7	1	0	0	14	%93
	Gerçek	1	1	0	4	7	1	0	0		
‘YKBNK.IS’	Model	2	3	4	2	10	2	0	0	23	%100
	Gerçek	2	3	4	2	10	2	0	0		
‘ZEDUR.IS’	Model	1	1	1	3	4	2	0	0	12	%100
	Gerçek	1	1	1	3	4	2	0	0		

Tablo 5’te 42 hisse senedinin tamamı için hem gerçek değerler hem de CNN modelinden elde edilen model sonuçları yer almaktadır. Tablonun içerisinde yer alan sayılar hisse senetlerinin hem model hem de gerçekte ilgili mum formasyonlarından kaçar tane elde edildiğini belirtmektedir. Tablo 5 incelendiğinde oluşturulan CNN modelinin oldukça başarılı sonuçlar verdiği görülmektedir. 42 hisse senedinin %52’sinde %100 başarı oranı yakalanmıştır. Geriye kalanlarda ise başarı oranı %81 ile %95 arasında değişim göstermektedir. Ortaya çıkan formasyon sayıları incelendiğinde diğer hisse senetlerine kıyasla hem daha az sayıda hem de daha çok sayıda formasyon sayılarının bulunduğu hisse senetlerinde yüksek başarı oranları elde edilmiştir. En düşük başarı oranına sahip hisse senedi %81 başarı oranı ile ‘TTKOM.IS’ olmuştur. Bu hisse senedinde 16 adet mum formasyonu ortaya çıkmıştır. Bunlardan 8 tanesi üç dıştan aşağı formasyonu, 7 tanesi üç içeride yukarı formasyonu ve 1 tanesi de üç dıştan yukarı formasyonudur. Bu formasyonlar arasında üç dıştan aşağı formasyonlarının tamamı model tarafından doğru sınıflandırılmıştır. Yanlış sınıflandırılan üç formasyonun tamamı gerçekte üç içeride yukarı formasyonu olması gerekirken üç dıştan yukarı formasyonu olarak sınıflandırılmıştır. Bu formasyonlar yanlış sınıflandırılmasına rağmen yükselen piyasa bilgisini doğru yakalayabilmiştir. Bu da modelin en azından piyasanın hareket yönü ile ilgili yatırımcıya doğru bir fikir verebileceğinin bir göstergesidir.

Modelin hisse senedi bazındaki başarısı kadar formasyonlar özelindeki sınıflandırma başarısı da önem arz etmektedir. Bu sebeple, her bir formasyonun sınıflandırma yapısını görebilmek adına, model sonucu ile gerçek değerler arasındaki karmaşıklık matrisi elde edilmiş ve sonuçlar Tablo 6’da sunulmuştur.

Tablo 6
Formasyonlar için Karmaşıklık (Confusion) Matrisi

		Gerçek Değer								Toplam
		Akşam Yıldızı	Sabah Yıldızı	Üç İçeride Aşağı	Üç İçeride Yukarı	Üç Dıştan Aşağı	Üç Dıştan Yukarı	Üç Yıldız Aşağı	Üç Yıldız Yukarı	
Model Sonucu	Akşam Yıldızı	70	0	1	0	0	0	0	0	71
	Sabah Yıldızı	0	44	0	0	0	0	0	0	44
	Üç İçeride Aşağı	2	0	42	0	0	0	0	0	44
	Üç İçeride Yukarı	0	6	0	101	0	0	0	0	107
	Üç Dıştan Aşağı	0	0	0	0	279	0	0	0	279
	Üç Dıştan Yukarı	0	0	0	13	0	84	0	1	98
	Üç Yıldız Aşağı	0	0	0	0	1	0	1	0	2
	Üç Yıldız Yukarı	0	0	0	0	0	0	3	4	7
	Toplam	72	50	43	114	280	84	4	5	652
Doğruluk Oranı		%97	%88	%97,6	%86	%99,6	%100	%25	%80	%95,85

Tablo 6’da test verileri için CNN modelinden elde edilen değerler ile gerçek formasyonların karşılaştırılması yer almaktadır. Test verisinde ortaya çıkan 652 formasyonun doğru sınıflandırma oranı %95,85 olarak hesaplanmıştır. Test verisinin tamamı incelendiğinde en çok üç dıştan aşağı formasyonunun ortaya çıktığı, bunu üç içeride yukarı formasyonunun takip ettiği görülmektedir. Test verisinde en az sayıda ortaya çıkan iki formasyon ise üç yıldız aşağı/yukarı formasyonlarıdır.

Formasyonların sınıflandırma oranları incelendiğinde en düşük doğru sınıflandırma oranına sahip olan formasyonun üç yıldız aşağı olduğu görülmektedir. Ancak bu formasyonun test verisinde ortaya çıkış sayısı göz önüne alındığında yalnızca dört adet ortaya çıktığı ve bunlardan bir tanesinin doğru sınıflandırıldığı dikkat çekmektedir. En yüksek doğru sınıflandırma oranına sahip olan formasyon ise ortaya çıkan 84 formasyonun tümünün doğru sınıflandırıldığı üç dıştan yukarı formasyonudur. Ayrıca diğer formasyonlar incelendiğinde sınıflandırma oranının genel olarak yüksek olduğu görülmektedir.

Yanlış sınıflandırılan formasyonlar incelendiğinde sayısal olarak yanlış sınıflandırılan formasyonun üç içeride yukarı formasyonu olduğu ve burada yanlış sınıflandırmaların üç dıştan yukarı (13 adet) olarak ortaya çıktığı tespit edilmiştir. Bu iki formasyon da boğa formasyonları olup, piyasanın hareketi için aynı yönü ifade etmektedir. Dolayısıyla CNN modelinin, formasyonu yanlış sınıflandırdığında bile piyasanın hareket yönünü doğru yakalayabildiği söylenebilir. Yanlış sınıflandırılan diğer formasyonlara bakıldığında sayısal olarak az olmakla birlikte piyasanın hareket yönünün model tarafından genellikle doğru tespit edildiği görülmektedir. Bu durum modelin başarılı tahminler yaptığı ve yatırımcı için doğru yönlendirmeler sunabileceğini göstermektedir.

Bu formasyonların doğru tespit edilmesi yatırımcının alacağı kararlar için önemlidir. Akşam yıldızı formasyonu ortaya çıktığında düşüş eğiliminin başladığının güvenilir bir göstergesi olarak kabul edilir. Sabah yıldızı formasyonu ise boğa piyasasını işaret eden bir formasyondur. Fiyat hareketindeki düşme eğiliminin ardından oluşan sabah yıldızı, yükselme eğiliminin başladığının önemli bir göstergesidir. Üç içeride yukarı/aşağı formasyonları kısa vadeli fiyat hareketinin yönünün değişebileceğine dair bir uyarı olarak kullanılabilir. Bu formasyonlar sıklıkla oluşmakla birlikte her zaman da güvenilir sonuçlara yönlendirmeyebilir. Formasyonun kısa vadeli olması, zaman zaman hareketin yönü bakımından önemli değişikliklere neden olsa da, yeni yönde yalnızca küçük-orta ölçekli bir hareket sağlayabilir. Üç dıştan yukarı formasyonu boğa piyasalarındaki güveni artırarak satın alma sinyallerini tetikler. Üç dıştan aşağı formasyonu ise ayı piyasasındaki güveni artırarak satış sinyallerini tetiklemektedir. Üç yıldız yukarı/aşağı formasyonları nadir görülen ancak ortaya çıktığında fiyat hareket yönünün keskin bir biçimde tersine döneceğine işaret eden bir formasyondur. Tüm formasyonlar yatırımcıya oldukça önemli bilgiler vermekle birlikte yatırımcının alacağı kararda tek başına etkili değildir. Yatırımcı bu anlamda diğer teknik ek göstergeleri de dikkate alarak hareket biçimine yön vermelidir.

4.4.2 LSTM ve GRU Bulguları

CNN modeli ile formasyonlar tespit edilip sınıflandırıldıktan sonra, elde edilen sınıflandırma olasılıkları kullanılarak hisse senetlerinin kapanış fiyatlarını ve fiyatların

hareket yönünü tahmin etmek (forecasting) üzere LSTM ve GRU algoritmaları uygulanmıştır. Bu amaçla, yanlış sınıflandırmadan doğacak hataların önüne geçmek adına, CNN algoritmasında doğru sınıflandırma oranı en yüksek olan hisse senetlerinden 3 tanesi seçilmiştir. Bu hisse senetleri YKBNK, BRMEN ve SAHOL olarak belirlenmiştir. Tablo 5’te verilen bulgular incelendiğinde her üç hisse senedi de %100 başarı ile doğru sınıflandırılmıştır. Bu hisse senetlerinin 22.07.2005 - 22.07.2023 tarihleri arasındaki 4598 günlük verisi alınmıştır. YKBNK formasyon sayısı bakımından daha çok formasyona sahip bir hisse senedir. Bu hisse senedinin incelenen dönem için mum grafiği Şekil 68’de verilmiştir.



Şekil 68: YKBNK Hisse Senedi Mum Grafiği

Şekil 68 incelendiğinde, YKBNK hisse senedinin seçilen diğer iki hisse senedine göre orta genişlikte fiyat aralığına sahip olduğu görülmektedir. İncelenen ikinci hisse senedi BRMEN’de daha az sayıda formasyon oluştuğu görülmüştür. Bu hisse senedinin incelenen dönem için mum grafiği Şekil 69’da verilmiştir.



Şekil 69: BRMEN Hisse Senedi Mum Grafiği

Şekil 69’da BRMEN hisse senedinin daha dar bir fiyat aralığına sahip olduğu görülmektedir. Son hisse senedi olan SAHOL hisse senedi de formasyon bakımından daha çok formasyona sahipken, aldığı değerler bakımından farklılaşmaktadır. Bu hisse senedinin incelenen dönem için mum grafiği Şekil 70’te verilmiştir.



Şekil 70: SAHOL Hisse Senedi Mum Grafiği

Şekil 70'e göre, SAHOL hisse senedi diğer ikisine kıyasla fiyat değerleri bakımından daha geniş aralıkta değerler almaktadır.

Tasarlanan LSTM ve GRU mimarilerinin farklı özellikteki veri setlerine karşı tutarlı sonuçlar verip vermediğini incelemek amacıyla bu üç hisse senedi için analizler yapılmıştır. Seçilen hisse senetlerinin her biri için ayrı eğitim ve tahmin işlemleri gerçekleştirilmiştir. Her üç hisse senedi için metodoloji kısmında bahsedilen olasılık, fiyat ve olasılık_fiyat veri setleri olmak üzere üç veri seti oluşturulmuştur. Ardından katman sayılarının ve nöron sayılarının sabit tutulduğu ortak model yapısı uygulanmıştır. Ortak olarak kullanılan model yapısı LSTM ve GRU için Tablo 7'de verilmiştir.

Tablo 7
Kullanılan LSTM - GRU Mimarileri

LSTM Modeli	GRU Modeli
LSTM(250), LSTM(200), Dropout(0.1), LSTM(150), Dropout(0.1), LSTM(75), Dropout(0.1), Dense(100, activation='tanh'), Dropout(0.1), Dense(units=horizon)	GRU(250), GRU (200), Dropout(0.1), GRU (150), Dropout(0.1), GRU (75), Dropout(0.1), Dense(100, activation='tanh'), Dropout(0.1), Dense(units=horizon)

Üç hisse senedi için oluşturulan olasılık, fiyat ve olasılık_fiyat veri setleri için %80'i eğitim verisi, %20'si geçerlilik verisi olarak ayrılmış, test verisi olarak ise eğitim ve geçerlilik verisi dışında kalan bir sonraki 10 gün alınmış ve Tablo 7'de yer alan iki mimari kullanılmıştır.

Tablo 7 incelendiğinde, LSTM ve GRU modellerinde öncelikle 250 nöronlu ardından 200 nöronlu olmak üzere LSTM ve GRU katmanları oluşturulmuştur. Ardından %10'luk seyreltme katmanı eklenerek aşırı öğrenme sorununun önüne geçilmiştir. Sonrasında sırasıyla 150 ve 75 nöronlu katmanlar eklenmiş ve aralarına seyreltme katmanı eklenerek ağ yapısı derinleştirilmiştir. Son olarak 100 nöronlu yoğunluk katmanı ve hiperbolik tanjant aktivasyon fonksiyonu ile devam edilmiş bu aşamada da seyreltme kullanılmıştır. Aynı model yapıları üç veri seti için uygulanmıştır. Hisse senetlerine uygulanan model yapılarının aynı olması sebebiyle katmanlarda kullanılan parametre sayıları hisse senetlerinde farklılık göstermemektedir; ancak veri setine göre değişken sayısı farklı olduğundan olasılık, fiyat ve olasılık_fiyat verilerinde farklılık göstermektedir. Bu sebeple, çalışmaya dahil edilen üç hisse senedi için ayrı ayrı katman yapılarına ilişkin parametre çıktıları bilgisi verilmemiş, sadece veri türlerine göre değişiklik gösteren ortak sonuçlara yer verilmiştir. İlk olarak olasılık veri seti için model özeti ve katmanlarda kullanılan parametre sayıları Tablo 8'de verilmiştir.

Tablo 8
Olasılık Veri Seti için LSTM ve GRU Model Yapıları

LSTM Modeli			GRU Modeli		
Katman(tür)	Model Yapısı	Parametre	Katman(tür)	Model Yapısı	Parametre
Istm	(None, 90, 250)	260000	gru	(None, 90, 250)	195750
Istm	(None, 90, 200)	360800	gru	(None, 90, 200)	271200
dropout	(None, 90, 200)	0	dropout	(None, 90, 200)	0
Istm	(None, 90, 150)	210600	gru	(None, 90, 150)	158400
dropout	(None, 90, 150)	0	dropout	(None, 90, 150)	0
Istm	(None, 75)	67800	gru	(None, 75)	51075
dropout	(None, 75)	0	dropout	(None, 75)	0
dense	(None, 100)	7600	dense	(None, 100)	7600
dropout	(None, 100)	0	dropout	(None, 100)	0
dense	(None, 10)	1010	dense	(None, 10)	1010
Toplam parametre: 907810 (3.46 MB) Eğitilebilir parametre: 907810 (3.46 MB) Eğitilemeyen parametre: 0 (0.00 Byte)			Toplam parametre: 685035 (2.61 MB) Eğitilebilir parametre: 685035 (2.61 MB) Eğitilemeyen parametre: 0 (0.00 Byte)		

Tablo 8 incelendiğinde LSTM modelinde 907810 adet parametre kullanılmış olup bunların tümü eğitilebilir parametrelerdir. GRU modelinde ise tümü eğitilebilir olan 685035 parametre kullanılmıştır. Fiyat veri seti için model özeti ve katmanlarda kullanılan parametre sayıları Tablo 9’da verilmiştir.

Tablo 9
Fiyat Veri Seti için LSTM ve GRU Model Yapıları

LSTM Modeli			GRU Modeli		
Katman(tür)	Model Yapısı	Parametre	Katman(tür)	Model Yapısı	Parametre
Istm	(None, 90, 250)	256000	gru	(None, 90, 250)	192750
Istm	(None, 90, 200)	360800	gru	(None, 90, 200)	271200
dropout	(None, 90, 200)	0	dropout	(None, 90, 200)	0
Istm	(None, 90, 150)	210600	gru	(None, 90, 150)	158400
dropout	(None, 90, 150)	0	dropout	(None, 90, 150)	0
Istm	(None, 75)	67800	gru	(None, 75)	51075
dropout	(None, 75)	0	dropout	(None, 75)	0
dense	(None, 100)	7600	dense	(None, 100)	7600
dropout	(None, 100)	0	dropout	(None, 100)	0
dense	(None, 10)	1010	dense	(None, 10)	1010
Toplam parametre: 903810 (3.45 MB) Eğitilebilir parametre: 903810 (3.45 MB) Eğitilemeyen parametre: 0 (0.00 Byte)			Toplam parametre: 682035 (2.60 MB) Eğitilebilir parametre: 682035 (2.60 MB) Eğitilemeyen parametre: 0 (0.00 Byte)		

Tablo 9 incelendiğinde LSTM modelinde 907810 adet parametre kullanılmış ve bunların tümü eğitilebilir parametrelerdir. GRU modelinde ise 682035 parametrenin tümü eğitilebilir olarak kullanılmıştır. Olasılık_Fiyat veri seti için model özeti ve katmanlarda kullanılan parametre sayıları Tablo 10’da verilmiştir.

Tablo 10
Olasılık_Fiyat Veri Seti için LSTM ve GRU Model Yapıları

LSTM Modeli			GRU Modeli		
Katman(tür)	Model Yapısı	Parametre	Katman(tür)	Model Yapısı	Parametre
lstm	(None, 90, 250)	26400	gru	(None, 90, 250)	198750
lstm	(None, 90, 200)	360800	gru	(None, 90, 200)	271200
dropout	(None, 90, 200)	0	dropout	(None, 90, 200)	0
lstm	(None, 90, 150)	210600	gru	(None, 90, 150)	158400
dropout	(None, 90, 150)	0	dropout	(None, 90, 150)	0
lstm	(None, 75)	67800	gru	(None, 75)	510075
dropout	(None, 75)	0	dropout	(None, 75)	0
dense	(None, 100)	7600	dense	(None, 100)	7600
dropout	(None, 100)	0	dropout	(None, 100)	0
dense	(None, 10)	1010	dense	(None, 10)	1010
Toplam parametre: 911810 (3.48 MB)			Toplam parametre: 688035 (2.62 MB)		
Eğitilebilir parametre: 911810 (3.48 MB)			Eğitilebilir parametre: 688035 (2.62 MB)		
Eğitilemeyen parametre: 0 (0.00 Byte)			Eğitilemeyen parametre: 0 (0.00 Byte)		

Tablo 10 incelendiğinde LSTM modelinde 911810 adet parametre kullanılmış olup bunların tümü eğitilebilir parametrelerdir. GRU modelinde ise 688035 parametre kullanılmıştır ve tümü eğitilebilir parametrelerdir.

Tüm hisse senetlerinde (BRMEN, SAHOL, YKBNK) standart bir model yapısı kullanıldığından katman yapıları ve hesaplanan toplam parametre sayıları aynı olacaktır. Ancak her hisse senedi için ayrı eğitim süreçleri gerçekleştirildiğinden farklı ağırlıklar hesaplanarak işlem yapılmıştır. Başka bir ifade ile model yapıları ortaktır, ancak eğitim süreçleri farklıdır. Bu durum modellerin kendi dinamiklerine göre ilerlemesi bakımından önemli bir noktadır.

Tüm hisse senetlerinde üç veri seti için de modeller eğitilmiş ve oldukça düşük kayıp fonksiyon değerleri elde edilerek başarılı bir şekilde tamamlanmıştır. Daha sonra

hisse senetlerinin test verilerinde her iki algoritma da uygulanmıştır. İlk olarak YKBNK hisse senedinin test verisine ilişkin sonuçlar Tablo 11’de verilmiştir.

Tablo 11
YKBNK Model Sonuçları

	LSTM		GRU	
	RMSE	MAE	RMSE	MAE
Olasılık Verisi	0.80	0.74	2.44	2.38
Fiyat Verisi	3.28	3.25	3.51	3.49
Olasılık_Fiyat Verisi	0.44	0.37	1.92	1.87

Tablo 11’de yer alan performans metrikleri incelendiğinde hem RMSE hem de MAE kriterine göre en düşük sonucu veren algoritmanın her üç veri seti için de LSTM olduğu görülmektedir. Buna göre, YKBNK hisse senedi için kapanış fiyatları tahmininde LSTM algoritmasının GRU algoritmasından daha başarılı olduğu söylenebilir. Ayrıca veri setlerine göre sonuçlar incelendiğinde, her iki algoritmada da olasılık verisi fiyat verisinden daha düşük hata ile sonuç vermektedir. Ancak en iyi sonuçların her iki veri setinin birleşimi olan olasılık_fiyat modelinden elde edildiği tespit edilmiştir. Buradan hareketle, kapanış fiyatlarının tahmin edilmesinde mum formasyonlarının kullanılmasının, fiyat verileri ile tahmin yapmaktan daha başarılı bir alternatif olduğu görülmektedir. Bununla birlikte, hem fiyat hem mum formasyonu bilgisinin bir arada kullanımı fiyat tahmini için en etkili seçenek olarak ortaya çıkmaktadır. Bir başka ifadeyle, fiyat verilerine görüntü olasılıklarının bilgisi eklendiğinde daha başarılı tahmin sonuçları elde edilmektedir. Veri setleri için elde edilen bu bulgunun, LSTM’e göre daha az başarılı olsa da GRU algoritmasında da geçerli olduğu görülmektedir.

Hisse senetlerinin kapanış fiyatlarını tahmin etmek kadar fiyatların hareket yönünü de belirlemek önem arz etmektedir. Mum formasyonları daha çok fiyatların hareket yönü hakkında bilgi almak amacıyla kullanılmaktadır. Tüm veri setlerinde başarılı sonuçlar verdiğinden, fiyatların tahmin değerlerini ve hareket yönlerini incelemek amacıyla LSTM algoritmasına ait bulgular kullanılacaktır. YKBNK hisse senedi için kapanış fiyatlarının tahminleri ve bu tahmin sonuçlarına bağlı olarak elde edilen hareket yönleri Tablo 12’de yer almaktadır.

Tablo 12
YKBNK Tahmin ve Hareket Yönü Sonuçları

Birim No	Gerçek		Olasılık		Fiyat		Olasılık_Fiyat	
	Fiyat Değeri	Yön	Tahmin	Yön	Tahmin	Yön	Tahmin	Yön
1	12.99	-	13.46	-	10.65	-	13.33	-
2	13.62	1	13.18	0	10.49	0	13.27	0
3	13.79	1	13.19	1	10.36	0	13.39	1
4	14.27	1	13.05	0	10.39	1	13.44	1
5	13.91	0	12.82	0	10.16	0	13.36	0
6	13.82	0	12.89	1	10.20	1	13.29	0
7	12.90	0	12.61	0	10.27	1	13.35	1
8	13.22	1	12.35	0	9.82	0	13.13	0
9	13.19	0	12.32	0	9.88	1	13.20	1
10	12.90	0	12.24	0	9.82	0	13.08	0
%				%56		%33		%56

Tablo 12’de fiyatların gerçek değerleri ile birlikte tüm veri setlerinden elde edilen tahmin değerleri yer almaktadır. Ayrıca fiyatların hareket yönlerine farklı sütunlarda yer verilmiştir. Yön sütunlarında yer alan 1 değeri, fiyatın bir önceki güne göre arttığını; 0 değeri ise bir önceki güne göre azaldığını göstermektedir. En son satırda yer alan oranlar ise gerçek hareket yönlerinin ne kadarının doğru tahmin edildiğini göstermektedir. Tablo 12’de yer alan tahmin değerleri incelendiğinde, karşılaştırma kriterlerinden elde edilen bulgularla tutarlı olarak, gerçek fiyatlara en yakın tahminlerin olasılık_fiyat veri setinden elde edildiği görülmektedir. Tablo 12’de yer alan sonuçlar hareket yönü açısından incelendiğinde, gerçek fiyatların hareket yönünü daha iyi yakalayan modellerin olasılık bilgisinin yer aldığı veri setlerinden (olasılık ve olasılık_fiyat) elde edilen modeller olduğu görülmektedir. Buradan hareketle, formasyon görüntü bilgisinin fiyatın değeri kadar yönü hakkında da bilgi verici olduğu anlaşılmaktadır.

Bir sonraki aşamada BRMEN hisse senedinin test verilerinden elde edilen sonuçlar Tablo 13’te verilmiştir.

Tablo 13
BRMEN Model Sonuçları

	LSTM		GRU	
	RMSE	MAE	RMSE	MAE
Olasılık Verisi	1.86	1.85	2.98	2.97
Fiyat Verisi	0.58	0.55	2.50	2.49
Olasılık_Fiyat Verisi	0.56	0.54	2.68	2.67

Tablo 13’de yer alan performans metrikleri incelendiğinde her iki kritere göre en düşük sonucu veren algoritmanın tüm veri setleri için LSTM olduğu görülmektedir. Buna göre, BRMEN hisse senedi için kapanış fiyatları tahmininde LSTM algoritmasının GRU algoritmasından daha başarılı olduğu söylenebilir. LSTM sonuçları veri setlerine göre incelendiğinde, her iki veri setinin birleşimi olan olasılık_fiyat modelinin daha başarılı olduğu tespit edilmiştir. Buradan hareketle, hem fiyat hem mum formasyonu bilgisinin bir arada kullanımı fiyat tahmini için en etkili seçenek olarak ortaya çıkmaktadır. Bir başka ifadeyle, fiyat verilerine görüntü olasılıklarının bilgisi eklendiğinde daha başarılı tahmin sonuçları elde edilmektedir.

BRMEN hisse senedi için kapanış fiyatlarının tahminleri ve bu tahmin sonuçlarına bağlı olarak elde edilen hareket yönleri Tablo 14’te yer almaktadır.

Tablo 14
BRMEN Tahmin ve Hareket Yönü Sonuçları

Birim No	Gerçek		Olasılık		Fiyat		Olasılık_Fiyat	
	Fiyat Değeri	Yön	Tahmin	Yön	Tahmin	Yön	Tahmin	Yön
1	2.61	-	4.58	-	2.02	-	2.06	-
2	2.83	1	4.39	0	2.08	1	2.07	1
3	2.62	0	4.50	1	2.03	0	2.06	0
4	2.79	1	4.51	1	2.01	0	2.07	1
5	2.75	0	4.52	1	2.03	1	2.04	0
6	2.66	0	4.38	0	2.04	1	2.06	1
7	2.40	0	4.42	1	2.07	1	2.04	0
8	2.43	1	4.45	1	2.04	0	2.07	1
9	2.50	1	4.40	0	2.05	1	2.05	0
10	2.36	0	4.37	0	2.04	0	2.02	0
%				%44		%44		%78

Tablo 14’te fiyatların gerçek değerleri ile birlikte üç veri setinden elde edilen tahmin değerleri yer almaktadır. Ayrıca fiyatların hareket yönlerine farklı sütunlarda yer verilmiştir. Yön sütunlarında yer alan 1 değeri, fiyatın bir önceki güne göre arttığını; 0 değeri ise bir önceki güne göre azaldığını göstermektedir. En son satırda yer alan oranlar ise gerçek hareket yönlerinin ne kadarının doğru tahmin edildiğini göstermektedir. Tablo 14’te yer alan tahmin değerleri incelendiğinde, karşılaştırma kriterlerinden elde edilen bulgularla tutarlı olarak, gerçek fiyatlara en yakın tahminlerin olasılık_fiyat veri setinden elde edildiği görülmektedir. Tablo 14’te yer alan sonuçlar hareket yönü açısından incelendiğinde, gerçek fiyatların hareket yönünü en yüksek oranla yakalayan modelin olasılık_fiyat verisi olduğu görülmektedir. Buna göre, mum formasyonlarının görüntülerinden elde edilen olasılıkların fiyat bilgisine eklenmesi ile hem fiyatların kapanış değerini hem de hareket yönünü tahmin etmek için daha başarılı bir model elde edilmiştir. Bu noktada, formasyon görüntü bilgisinin fiyatın değeri kadar yönü hakkında da bilgi verici olduğu anlaşılmaktadır.

Son olarak SAHOL hisse senedinin test verilerinden elde edilen sonuçlara Tablo 15’te yer verilmiştir.

Tablo 15
SAHOL Model Sonuçları

	LSTM		GRU	
	RMSE	MAE	RMSE	MAE
Olasılık Verisi	6.44	4.98	6.87	5.66
Fiyat Verisi	5.62	5.34	9.77	9.39
Olasılık_Fiyat Verisi	4.99	4.45	7.63	6.87

Tablo 15’te yer alan performans metrikleri incelendiğinde her iki kritere göre en düşük sonucu veren algoritmanın tüm veri setleri için LSTM olduğu görülmektedir. Buna göre, SAHOL hisse senedi için kapanış fiyatları tahmininde LSTM algoritmasının GRU algoritmasından daha başarılı olduğu söylenebilir. LSTM sonuçları veri setlerine göre incelendiğinde, olasılık_fiyat modelinin daha başarılı olduğu görülmektedir. Buradan hareketle, hem fiyat bilgisinin hem de görüntü sınıflandırma olasılıklarının bir arada kullanımının fiyat tahmini için en uygun kullanım olduğu anlaşılmaktadır. SAHOL hisse

senedi için kapanış fiyatlarının tahminleri ve bu tahmin sonuçlarına bağlı olarak elde edilen hareket yönleri Tablo 16’da yer almaktadır.

Tablo 16
SAHOL Tahmin ve Hareket Yönü Sonuçları

Birim No	Gerçek		Olasılık		Fiyat		Olasılık_Fiyat	
	Fiyat Değeri	Yön	Tahmin	Yön	Tahmin	Yön	Tahmin	Yön
1	47.10	-	39.48	-	55.64	-	48.16	-
2	48.88	1	61.41	1	56.11	1	47.82	0
3	50.00	1	53.06	0	55.09	0	47.00	0
4	50.55	1	49.72	0	56.29	1	47.27	1
5	52.5	1	47.09	0	56.94	1	47.10	0
6	53.95	1	52.28	1	56.20	0	47.23	1
7	50.55	0	51.95	0	56.71	1	46.93	0
8	52.35	1	40.94	0	58.54	1	46.27	0
9	52.75	1	51.85	1	56.00	0	46.52	1
10	52.40	0	47.43	0	56.95	1	44.38	0
%				%56		%44		%56

Tablo 16’da fiyatların gerçek değerleri ile birlikte tüm veri setlerinden elde edilen tahmin değerleri yer almaktadır. Ayrıca fiyatların hareket yönlerine farklı sütunlarda yer verilmiştir. Yön sütunlarında yer alan 1 değeri, fiyatın bir önceki güne göre arttığını; 0 değeri ise bir önceki güne göre azaldığını göstermektedir. En son satırda yer alan oranlar ise gerçek hareket yönlerinin ne kadarının doğru tahmin edildiğini göstermektedir. Tablo 16’da yer alan tahmin değerleri incelendiğinde, karşılaştırma kriterlerinden elde edilen bulgularla tutarlı olarak, gerçek fiyatlara en yakın tahminlerin olasılık_fiyat veri setinden elde edildiği görülmektedir. Tablo 16’da yer alan sonuçlar hareket yönü açısından incelendiğinde, gerçek fiyatların hareket yönünü daha iyi yakalayan modellerin olasılık bilgisinin yer aldığı veri setlerinden (olasılık ve olasılık_fiyat) elde edilen modeller olduğu görülmektedir. Böylece formasyon görüntü bilgisinin fiyatın değeri kadar yönü hakkında da bilgi verici olduğu anlaşılmaktadır.

Bulgular genel olarak incelendiğinde, hisse senetleri için kapanış fiyatlarının tahmininde LSTM algoritmasının daha başarılı olduğu görülmüştür. Ayrıca mum formasyonlarından elde edilen görüntü bilgisinin, modellerin tahmin becerisini

iyileřtirdiđi tespit edilmiřtir. Fiyat tahmininde en etkili sonular, grnt sınıflandırma olasılıklarının fiyat bilgileri ile birlikte kullanıldıđı modellerden elde edilmektedir. Benzer durum fiyat hareketlerinin yn iin de geerlidir. Hareketlerin ynn tahmin etmede en bařarılı modellerin grnt olasılıklarının yer aldıđı modeller olduđu belirlenmiřtir. Farklı formasyon sayıları ve farklı deđer aralıkları olan hisse senetleri iin elde edilen sonular birbiriyle tutarlıdır. Buna gre, nerilen olasılık temelli derin đrenme modeli, ortaya ıkan formasyon sayısının az ya da ok olması veya hisse senedinin deđer aralıđının geniř veya dar olması durumlarında fiyat ve yn tahmini iin kullanılabilir.



5. SONUÇ

Finansal sektör uygulamalarında yatırımcıların piyasa hareketlerini inceleyerek kâr elde edebilecek dinamikleri keşfetmeleri hedeflenmektedir. Bir piyasanın etkin olduğu durumda, mevcut piyasa bilgisinin piyasadaki tüm aktörler tarafından ulaşılabilir olduğu varsayılmaktadır. Ancak piyasalar genel olarak etkin olmadığı için mevcut piyasa bilgisi asimetrik olarak dağılmaktadır. Dolayısıyla doğru yatırım kararlarının alınabilmesi için piyasa bilgisini elde etmek ve var olan bilgiden ileriye yönelik çıkarımlar yapmak önem arz etmektedir. Bu durum yatırımcıların, yatırım ortamlarını iyi tanımalarını ve piyasayı iyi analiz edebilmelerini gerektirmektedir. Bu doğrultuda, makroekonomik göstergelerin baz alındığı temel analiz ile geçmiş dönemlerdeki fiyat ve hacim değişimlerinden yola çıkarak bir menkul değer fiyatı hakkında sonuç çıkarımı yapan teknik analiz yöntemleri kullanılmaktadır.

Temel analizde, hem sektör hem şirket dinamikleri hem de ülkelerin makro göstergelerinin birlikte değerlendirilmesiyle yatırım yapılacak menkul değer inceleme yapılmaktadır. Ancak bu sürecin oldukça donanım isteyen bir süreç olması sebebiyle alanında uzman kişiler tarafından gerçekleştirilmesi gerekmektedir. Teknik analizde ise, sektörel ve makroekonomik göstergelere doğrudan ihtiyaç duyulmamaktadır. Bu bilgilerin geçmiş dönem fiyatlarında dolaylı olarak yansıtıldığı varsayımıyla hareket edilmektedir. Dolayısıyla bu analizde geçmiş dönem fiyatlarından gelecek döneme dair tahminler yapılmakta ve bu tahminlere göre yatırım kararları alınmaktadır. Bu iki analizin birbirine göre bir üstünlüğü bulunmamakla birlikte, uzman kişiler tarafından yapıldığında ikisinden de başarılı sonuçlar elde edilmektedir.

Yatırımcılar finansal davranış biçimlerini teknik analizle belirlemek istediklerinde, bazı indikatörlerin ve grafiklerin ne anlama geldiğini iyi okumaları gerekir. Alınan zaman frekansına (günlük, aylık, yıllık vb.) göre fiyatların açılış, kapanış, en düşük, en yüksek değerleri indikatörlerin ve grafiklerin elde edilmesinde kullanılan değerlerdir. Fiyat hareketliliklerinden yola çıkarak yatırımcının doğru karar mekanizmaları geliştirmesi için kullanılan grafiklerden biri mum grafiğidir. Mum grafikleri, hisse senetleri için belirtilen fiyat değerlerinin bir arada bulunması bakımından

bilgi verici grafiklerdir. Öte yandan, bu grafiklerde ortaya çıkan ve tekrar eden benzer yapıları içeren mum formasyonları, hisse senetlerine ait fiyatların genel gidişatını incelemek açısından kullanışlıdır. Çünkü bu formasyonlar hisse senedindeki genel eğilimin devamlılığını ya da dönüşümünü göstermektedir. Bu bakımdan ayı/boğa piyasası geçişleri bu formasyonlarla takip edilebilmektedir.

Finansal uygulamalarda kullanılan geleneksel yöntemlerin yerine son dönemlerde derin öğrenme yöntemlerine sıklıkla başvurulmaktadır. Veriden öğrenme altyapısına dayalı derin öğrenme algoritmaları, veri çeşitliliği ve hacminin artmasıyla birlikte önem kazanmıştır. Bu algoritmalar, sayısal verilerin yanı sıra görüntü, ses, video, metin vb. veri türlerinin analiz edilmesine olanak sağlamaktadır. Ayrıca bilgi teknolojilerindeki hızlı gelişmelere paralel olarak ortaya çıkan büyük ve yüksek boyutlu veriler üzerinde çalışılması mümkün hale gelmiştir. Bu tür veriler için geleneksel yöntemler zayıf kalmakta, hatta bazı durumlarda kullanılmaları mümkün olmamaktadır. Bu noktada güçlü bir alternatif olarak sinir ağlarını temel alan algoritmalar geliştirilmiştir. Sinir ağlarındaki öğrenme yapısı dikkate alınarak karmaşıklık arttıkça katman sayıları artmakta ve derin öğrenme algoritmaları ortaya çıkmaktadır. Ayrıca verinin türüne göre konvolüsyon ve hafıza katmanları gibi özel yapılar eklenerek algoritmalar çeşitlenmektedir. Bu algoritmalar nesne tanıma, görüntü sınıflandırma, tahmin, öngörü, doğal dil işleme gibi amaçlar için kullanılmaktadır.

Bu çalışmada, derin öğrenme algoritmalarından CNN, LSTM ve GRU kullanılarak BİST30 hisse senetleri için ileriye dönük fiyat hareketliliği tahmini yapılmıştır. Bu amaçla hisse senetlerinin mum grafiklerinden elde edilen mum formasyonlarına ait görüntüler kullanılmıştır. Teknik analizde ayı ve boğa piyasalarına ait çok sayıda mum formasyonu olmakla birlikte, üçlü mum formasyonlarının fiyatların gidişatındaki dönüşümü iyi temsil ettiği bilinmektedir. Bu sebeple çalışmada üçlü formasyonlardan BİST30 hisse senetlerinde en çok gözlenen 8 mum formasyonu ele alınmıştır. Bu formasyonlar boğa piyasasına ait sabah yıldızı, üç içeride yukarı, üç dıştan yukarı ve üç yıldız yukarı; ayı piyasasından akşam yıldızı, üç içeride aşağı, üç dıştan aşağı ve üç yıldız aşağı formasyonlarıdır. Bu formasyonların doğru tespit edilmesi yatırımcının piyasa hareketine göre kendi davranış biçimini belirlemesi bakımından önemlidir. Bu

nedenle öncelikle formasyonlar için CNN algoritması ile görüntü sınıflandırma modeli geliştirilmiştir. Ancak CNN algoritmasının kullanılması etiketli verinin varlığını gerektirmektedir. Bu noktada hisse senetlerinin fiyat değerlerinden hareketle üçlü mum grafikleri çizdirilmiş ve python programında yer alan TA-Lib kütüphanesi yardımıyla etiketleme işlemi gerçekleştirilmiştir. Böylece formasyonlar için hazır bir veri seti olarak bulunmayan etiketli verilerin elde edilmesi otomatik olarak yapılmış ve araştırmacı hatalarının önüne geçilmiştir.

Formasyon görüntülerinin sınıflandırılması için CNN algoritması kullanılarak başarılı bir sınıflandırma modeli oluşturulmuştur. CNN algoritmasının temel katmanlarına ek olarak aşırı öğrenme sorununa önlem almak amacıyla seyreltme katmanı kullanılmıştır. Hisse senetlerine ilişkin test verileri incelendiğinde modelin, her hisse senedi için doğru sınıflandırma oranlarının yüksek olduğu görülmüştür. Burada yapılan sınıflandırma işlemi, yatırımcıların alacağı kararlar için formasyonların doğru tespit edilmesi açısından önemlidir. Fiyatların düşme eğiliminde olduğu durumda, boğa piyasasına ait formasyonların tespiti, kısa vadede bu eğilimin değişeceğini ve artışa geçeceğini göstermektedir. Benzer şekilde artış eğilimindeki fiyatlar için, ayı piyasasına ait formasyonların belirlenmesi, kısa vadede bu eğilimin değişerek düşüşe geçeceğinin sinyalini vermektedir. Bu bakımdan formasyonlar önceden doğru bir şekilde tespit edildiğinde yatırımcının finansal davranış biçimini geliştirmesinde önemli fikirler verebilir. Çalışmada geliştirilen CNN modeli bu açıdan kullanışlı bir modeldir.

Formasyonlar kısa vadede hareket yönündeki önemli değişikliklerin sinyallerini içermeleri bakımından bilgi verici olsa da, fiyatlarda ortaya çıkan değişiklikler küçük veya büyük ölçekli olabilir. Bu nedenle hareket yönünü öngörebilmek kadar fiyatların gelecek dönemlerini tahmin etmek de önemlidir. Çalışmanın bir sonraki aşamasında fiyatların gelecek dönem değerlerinin öngörüsünü yapmak ve hareket yönlerini belirlemek için LSTM ve GRU algoritmaları ile öngörü modelleri geliştirilmiştir. Burada mum formasyon görüntülerinin sınıflandırma bilgisi modele dahil edilerek, bu bilginin öngörü başarısı üzerindeki etkisi incelenmiştir. Görüntü sınıflandırma bilgisi, özgün istatistiksel bir yaklaşımla, her bir formasyona ait sınıflandırma olasılıkları olarak alınmıştır. Bu amaçla üç veri seti oluşturulmuştur. Bunlardan ilki sadece sınıflandırma

olasılıklarının yer aldığı veri seti, ikincisi sadece fiyat değerleri ve işlem hacminin yer aldığı veri setidir. Üçüncüsü ise her iki veri setinin birleştirilmesiyle oluşturulmuştur. Model bulguları incelendiğinde, her üç veri setinde de LSTM modelinin fiyat tahmininde daha başarılı olduğu tespit edilmiştir. Veri setleri açısından bakıldığında, olasılıkların fiyat bilgilerine eklenerek oluşturulan veri setinin tekli veri setlerine göre daha düşük hata oranına sahip olduğu görülmüştür. Buna göre, fiyat bilgilerine ek olarak formasyon bilgilerinin de kullanılması daha iyi fiyat tahminleri yapılmasını sağlamaktadır.

Çalışmada fiyatların gelecek dönem değerleri ile birlikte gelecek dönemdeki hareket yönleri de tahmin edilmiştir. Hisse senetlerinin hareket yönleri, fiyat tahmininde başarılı bulunan LSTM modelinin sonuçlarına göre belirlendiğinde, olasılık bilgisinin yer aldığı veri setlerinde daha yüksek oranda doğru hareket yönü tespit edilmiştir. Buna göre, formasyonların bilgisi, fiyatın değeri kadar, hareket yönünü tahmin etmede de avantaj sağlamaktadır. Geliştirilen LSTM modelinin her iki amaçla da kullanılabileceği anlaşılmaktadır.

Sonuç olarak, derin öğrenme algoritmalarının hisse senetlerinin fiyat hareketliliğinin öngörüsünde başarılı bir şekilde kullanılabileceği tespit edilmiştir. Geleneksel yöntemlerde sıklıkla kullanılan mum formasyonları görüntü verisi olarak ele alındığında da doğru oluşturulan modellerle oldukça bilgi verici olmaktadır. Bu bakımdan finans sektöründe farklı türde verilerin analiz edilmesi için derin öğrenme yöntemleri başarılı bir şekilde kullanılabilir. Böylece yeni türde veriler bir araya getirilerek oluşturulan veri setleri ile alternatif yatırım modelleri geliştirilebilir.

Zayıf etkinlik düzeyindeki piyasalarda spekülatif hareketlerden kaynaklı olarak daha sık fiyat hareketlilikleri yaşanabilmektedir. Formasyonlar, bu hareketliliklerin kısa vadede nispeten tespit edilebilmesini sağlamaktadır. Çalışmada, literatürde çok sayıda bulunan formasyonlardan kısıtlı sayıda kullanılması durumunda bile fiyat tahmininde etkili olduğu görülmüştür. Daha sonraki çalışmalarda bu formasyonların sayısı ve çeşitliliği artırılabilir veya indikatörler, makroekonomik ve sektörel göstergeler, piyasalara yön veren açıklamaların yer aldığı metin verileri vb. alternatif değişkenler kullanılabilir. Böylece derin öğrenme algoritmaları yardımıyla daha kapsamlı yatırım modelleri geliştirilebilir.



EKLER

EK 1. BİST 30 Hisse Senetlerinin CNN Sonuçları

ARCLK	Model Sonucu	Gerçek Değer
1	Üç İçeride Yukarı	Üç İçeride Yukarı
2	Üç İçeride Aşağı	Üç İçeride Aşağı
3	Üç Dıştan Aşağı	Üç Dıştan Aşağı
4	Üç Dıştan Yukarı	Üç Dıştan Yukarı
5	Üç Dıştan Aşağı	Üç Dıştan Aşağı
6	Üç Dıştan Yukarı	Üç Dıştan Yukarı
7	Üç Dıştan Aşağı	Üç Dıştan Aşağı
8	Akşam Yıldızı	Akşam Yıldızı
9	Akşam Yıldızı	Akşam Yıldızı
10	Akşam Yıldızı	Akşam Yıldızı
11	Akşam Yıldızı	Akşam Yıldızı
12	Akşam Yıldızı	Akşam Yıldızı
13	Sabah Yıldızı	Sabah Yıldızı
14	Sabah Yıldızı	Sabah Yıldızı
15	Sabah Yıldızı	Sabah Yıldızı
16	Üç Yıldız Aşağı	Üç Yıldız Aşağı
17	Üç Yıldız Yukarı	Üç Yıldız Yukarı

ASELS	Model Sonucu	Gerçek Değer
1	Üç İçeride Yukarı	Üç İçeride Yukarı
2	Üç İçeride Yukarı	Üç İçeride Yukarı
3	Üç İçeride Aşağı	Üç İçeride Aşağı
4	Üç İçeride Yukarı	Üç İçeride Yukarı
5	Üç Dıştan Aşağı	Üç Dıştan Aşağı
6	Üç Dıştan Aşağı	Üç Dıştan Aşağı
7	Üç Dıştan Aşağı	Üç Dıştan Aşağı
8	Üç Dıştan Aşağı	Üç Dıştan Aşağı
9	Üç Dıştan Aşağı	Üç Dıştan Aşağı
10	Üç Dıştan Yukarı	Üç Dıştan Yukarı
11	Üç Dıştan Aşağı	Üç Dıştan Aşağı
12	Üç Dıştan Aşağı	Üç Dıştan Aşağı
13	Akşam Yıldızı	Akşam Yıldızı
14	Sabah Yıldızı	Sabah Yıldızı
15	Sabah Yıldızı	Sabah Yıldızı

BIMAS	Model Sonucu	Gerçek Değer
1	Üç İçeride Yukarı	Üç İçeride Yukarı
2	Üç İçeride Yukarı	Üç İçeride Yukarı
3	Üç İçeride Yukarı	Üç İçeride Yukarı
4	Üç İçeride Yukarı	Üç İçeride Yukarı
5	Üç Dıştan Yukarı	Üç Dıştan Yukarı
6	Üç Dıştan Aşağı	Üç Dıştan Aşağı
7	Üç Dıştan Aşağı	Üç Dıştan Aşağı
8	Üç Dıştan Aşağı	Üç Dıştan Aşağı
9	Üç Dıştan Aşağı	Üç Dıştan Aşağı
10	Üç Dıştan Yukarı	Üç Dıştan Yukarı
11	Akşam Yıldızı	Akşam Yıldızı
12	Sabah Yıldızı	Sabah Yıldızı
13	Sabah Yıldızı	Sabah Yıldızı
14	Sabah Yıldızı	Sabah Yıldızı
15	Sabah Yıldızı	Sabah Yıldızı

BRMEN	Model Sonucu	Gerçek Değer
1	Üç İçeride Yukarı	Üç İçeride Yukarı
2	Üç İçeride Yukarı	Üç İçeride Yukarı
3	Üç İçeride Yukarı	Üç İçeride Yukarı
4	Üç İçeride Yukarı	Üç İçeride Yukarı
5	Üç Dıştan Aşağı	Üç Dıştan Aşağı
6	Üç Dıştan Aşağı	Üç Dıştan Aşağı
7	Üç Dıştan Aşağı	Üç Dıştan Aşağı
8	Üç Dıştan Aşağı	Üç Dıştan Aşağı
9	Sabah Yıldızı	Sabah Yıldızı

CMEN	Model Sonucu	Gerçek Değer
1	Üç İçeride Yukarı	Üç İçeride Yukarı
2	Üç İçeride Aşağı	Üç İçeride Aşağı
3	Üç İçeride Yukarı	Üç İçeride Yukarı
4	Üç Dıştan Yukarı	Üç Dıştan Yukarı
5	Üç Dıştan Aşağı	Üç Dıştan Aşağı
6	Üç Dıştan Aşağı	Üç Dıştan Aşağı
7	Üç Dıştan Yukarı	Üç Dıştan Yukarı
8	Üç Dıştan Aşağı	Üç Dıştan Aşağı
9	Üç Dıştan Aşağı	Üç Dıştan Aşağı
10	Akşam Yıldızı	Akşam Yıldızı
11	Sabah Yıldızı	Sabah Yıldızı

CMENT	Model Sonucu	Gerçek Değer
12	Sabah Yıldızı	Sabah Yıldızı
13	Sabah Yıldızı	Sabah Yıldızı
14	Üç Yıldız Yukarı	Üç Yıldız Yukarı

DENGE	Model Sonucu	Gerçek Değer
1	Üç Dıştan Yukarı	Üç Dıştan Yukarı
2	Üç Dıştan Aşağı	Üç Dıştan Aşağı
3	Üç Dıştan Aşağı	Üç Dıştan Aşağı
4	Üç Dıştan Aşağı	Üç Dıştan Aşağı
5	Üç Dıştan Aşağı	Üç Dıştan Aşağı
6	Akşam Yıldızı	Akşam Yıldızı
7	Sabah Yıldızı	Sabah Yıldızı

EKGYO	Model Sonucu	Gerçek Değer
1	Üç İçeride Yukarı	Üç İçeride Yukarı
2	Üç İçeride Yukarı	Üç İçeride Yukarı
3	Üç Dıştan Aşağı	Üç Dıştan Aşağı
4	Üç Dıştan Aşağı	Üç Dıştan Aşağı
5	Üç Dıştan Aşağı	Üç Dıştan Aşağı
6	Üç Dıştan Yukarı	Üç Dıştan Yukarı
7	Üç Dıştan Aşağı	Üç Dıştan Aşağı
8	Üç Dıştan Aşağı	Üç Dıştan Aşağı
9	Üç Dıştan Aşağı	Üç Dıştan Aşağı
10	Üç Dıştan Aşağı	Üç Dıştan Aşağı
11	Üç Dıştan Yukarı	Üç Dıştan Yukarı
12	Üç Dıştan Aşağı	Üç Dıştan Aşağı
13	Akşam Yıldızı	Akşam Yıldızı
14	Akşam Yıldızı	Akşam Yıldızı
15	Akşam Yıldızı	Akşam Yıldızı

EREGL	Model Sonucu	Gerçek Değer
1	Üç İçeride Yukarı	Üç İçeride Yukarı
2	Üç İçeride Yukarı	Üç İçeride Yukarı
3	Üç Dıştan Aşağı	Üç Dıştan Aşağı
4	Üç Dıştan Aşağı	Üç Dıştan Aşağı
5	Üç Dıştan Aşağı	Üç Dıştan Aşağı
6	Üç Dıştan Aşağı	Üç Dıştan Aşağı
7	Üç Dıştan Aşağı	Üç Dıştan Aşağı
8	Üç Dıştan Aşağı	Üç Dıştan Aşağı

EREGL	Model Sonucu	Gerçek Değer
9	Üç Dıştan Aşağı	Üç Dıştan Aşağı
10	Üç Dıştan Yukarı	Üç Dıştan Yukarı
11	Üç Dıştan Aşağı	Üç Dıştan Aşağı
12	Üç Dıştan Aşağı	Üç Dıştan Aşağı
13	Üç Dıştan Aşağı	Üç Dıştan Aşağı
14	Üç Dıştan Yukarı	Üç Dıştan Yukarı
15	Akşam Yıldızı	Akşam Yıldızı
16	Akşam Yıldızı	Akşam Yıldızı
17	Akşam Yıldızı	Akşam Yıldızı
18	Sabah Yıldızı	Sabah Yıldızı

FROTO	Model Sonucu	Gerçek Değer
1	Üç İçeride Yukarı	Üç İçeride Yukarı
2	Üç Dıştan Yukarı	Üç İçeride Yukarı
3	Üç İçeride Yukarı	Üç İçeride Yukarı
4	Üç İçeride Aşağı	Üç İçeride Aşağı
5	Üç Dıştan Aşağı	Üç Dıştan Aşağı
6	Üç Dıştan Aşağı	Üç Dıştan Aşağı
7	Üç Dıştan Aşağı	Üç Dıştan Aşağı
8	Üç Dıştan Aşağı	Üç Dıştan Aşağı
9	Akşam Yıldızı	Akşam Yıldızı
10	Akşam Yıldızı	Akşam Yıldızı
11	Sabah Yıldızı	Sabah Yıldızı
12	Sabah Yıldızı	Sabah Yıldızı
13	Sabah Yıldızı	Sabah Yıldızı
14	Üç Yıldız Yukarı	Üç Yıldız Yukarı

GARAN	Model Sonucu	Gerçek Değer
1	Üç İçeride Aşağı	Üç İçeride Aşağı
2	Üç İçeride Yukarı	Üç İçeride Yukarı
3	Üç Dıştan Aşağı	Üç Dıştan Aşağı
4	Üç Dıştan Yukarı	Üç Dıştan Yukarı
5	Üç Dıştan Yukarı	Üç Dıştan Yukarı
6	Üç Dıştan Aşağı	Üç Dıştan Aşağı
7	Üç Dıştan Aşağı	Üç Dıştan Aşağı
8	Üç Dıştan Aşağı	Üç Dıştan Aşağı
9	Üç Dıştan Yukarı	Üç Dıştan Yukarı
10	Üç Dıştan Aşağı	Üç Dıştan Aşağı
11	Üç Dıştan Yukarı	Üç Dıştan Yukarı

GARAN	Model Sonucu	Gerçek Değer
12	Üç Dıştan Yukarı	Üç Dıştan Yukarı
13	Üç Dıştan Aşağı	Üç Dıştan Aşağı
14	Üç Dıştan Aşağı	Üç Dıştan Aşağı
15	Akşam Yıldızı	Akşam Yıldızı
16	Akşam Yıldızı	Akşam Yıldızı
17	Akşam Yıldızı	Akşam Yıldızı
18	Üç Yıldız Yukarı	Üç Yıldız Aşağı
19	Üç Dıştan Yukarı	Üç Yıldız Yukarı

GUBRF	Model Sonucu	Gerçek Değer
1	Üç İçeride Yukarı	Üç İçeride Yukarı
2	Üç İçeride Yukarı	Üç İçeride Yukarı
3	Üç İçeride Yukarı	Üç İçeride Yukarı
4	Üç İçeride Yukarı	Üç İçeride Yukarı
5	Üç İçeride Yukarı	Üç İçeride Yukarı
6	Üç Dıştan Aşağı	Üç Dıştan Aşağı
7	Üç Dıştan Aşağı	Üç Dıştan Aşağı
8	Üç Dıştan Aşağı	Üç Dıştan Aşağı
9	Üç Dıştan Yukarı	Üç Dıştan Yukarı
10	Üç Dıştan Aşağı	Üç Dıştan Aşağı
11	Üç Dıştan Aşağı	Üç Dıştan Aşağı
12	Üç Dıştan Yukarı	Üç Dıştan Yukarı
13	Üç Dıştan Aşağı	Üç Dıştan Aşağı
14	Üç Dıştan Aşağı	Üç Dıştan Aşağı
15	Üç Dıştan Aşağı	Üç Dıştan Aşağı
16	Üç Dıştan Yukarı	Üç Dıştan Yukarı
17	Üç Dıştan Aşağı	Üç Dıştan Aşağı
18	Üç Dıştan Yukarı	Üç Dıştan Yukarı
19	Üç Dıştan Aşağı	Üç Dıştan Aşağı
20	Akşam Yıldızı	Akşam Yıldızı
21	Akşam Yıldızı	Akşam Yıldızı
22	Üç İçeride Yukarı	Sabah Yıldızı

HALKB	Model Sonucu	Gerçek Değer
1	Üç İçeride Yukarı	Üç İçeride Yukarı
2	Üç İçeride Aşağı	Üç İçeride Aşağı
3	Üç Dıştan Yukarı	Üç İçeride Yukarı
4	Üç Dıştan Yukarı	Üç Dıştan Yukarı
5	Üç Dıştan Yukarı	Üç Dıştan Yukarı

HALKB	Model Sonucu	Gerçek Değer
6	Üç Dıştan Aşağı	Üç Dıştan Aşağı
7	Üç Dıştan Aşağı	Üç Dıştan Aşağı
8	Üç Dıştan Aşağı	Üç Dıştan Aşağı
9	Üç Dıştan Aşağı	Üç Dıştan Aşağı
10	Üç Dıştan Aşağı	Üç Dıştan Aşağı
11	Üç Dıştan Aşağı	Üç Dıştan Aşağı
12	Üç Dıştan Aşağı	Üç Dıştan Aşağı
13	Üç Dıştan Yukarı	Üç Dıştan Yukarı

HEKTS	Model Sonucu	Gerçek Değer
1	Üç İçeride Aşağı	Üç İçeride Aşağı
2	Üç İçeride Yukarı	Üç İçeride Yukarı
3	Üç İçeride Aşağı	Üç İçeride Aşağı
4	Üç İçeride Aşağı	Üç İçeride Aşağı
5	Üç İçeride Aşağı	Üç İçeride Aşağı
6	Üç Dıştan Aşağı	Üç Dıştan Aşağı
7	Üç Dıştan Aşağı	Üç Dıştan Aşağı
8	Üç Dıştan Aşağı	Üç Dıştan Aşağı
9	Üç Dıştan Yukarı	Üç Dıştan Yukarı
10	Üç Dıştan Yukarı	Üç Dıştan Yukarı
11	Üç Dıştan Aşağı	Üç Dıştan Aşağı
12	Üç Dıştan Aşağı	Üç Dıştan Aşağı
13	Üç Dıştan Aşağı	Üç Dıştan Aşağı
14	Üç Dıştan Aşağı	Üç Dıştan Aşağı
15	Üç Dıştan Aşağı	Üç Dıştan Aşağı
16	Üç Dıştan Aşağı	Üç Dıştan Aşağı
17	Akşam Yıldızı	Akşam Yıldızı
18	Akşam Yıldızı	Akşam Yıldızı
19	Akşam Yıldızı	Akşam Yıldızı

ISBIR	Model Sonucu	Gerçek Değer
1	Üç İçeride Yukarı	Üç İçeride Yukarı
2	Akşam Yıldızı	Üç İçeride Aşağı
3	Üç İçeride Aşağı	Üç İçeride Aşağı
4	Üç İçeride Yukarı	Üç İçeride Yukarı
5	Üç Dıştan Aşağı	Üç Dıştan Aşağı
6	Üç Dıştan Aşağı	Üç Dıştan Aşağı
7	Üç Dıştan Aşağı	Üç Dıştan Aşağı
8	Üç Dıştan Yukarı	Üç Dıştan Yukarı

ISBIR	Model Sonucu	Gerçek Değer
9	Üç Dıştan Aşağı	Üç Dıştan Aşağı
10	Üç Dıştan Yukarı	Üç Dıştan Yukarı
11	Üç Dıştan Aşağı	Üç Dıştan Aşağı
12	Üç Dıştan Aşağı	Üç Dıştan Aşağı
13	Akşam Yıldızı	Akşam Yıldızı
14	Akşam Yıldızı	Akşam Yıldızı
15	Akşam Yıldızı	Akşam Yıldızı

ISCTR	Model Sonucu	Gerçek Değer
1	Üç İçeride Yukarı	Üç İçeride Yukarı
2	Üç İçeride Yukarı	Üç İçeride Yukarı
3	Üç İçeride Yukarı	Üç İçeride Yukarı
4	Üç İçeride Yukarı	Üç İçeride Yukarı
5	Üç İçeride Aşağı	Üç İçeride Aşağı
6	Üç Dıştan Yukarı	Üç İçeride Yukarı
7	Üç İçeride Yukarı	Üç İçeride Yukarı
8	Üç İçeride Yukarı	Üç İçeride Yukarı
9	Üç Dıştan Yukarı	Üç Dıştan Yukarı
10	Üç Dıştan Aşağı	Üç Dıştan Aşağı
11	Üç Dıştan Aşağı	Üç Dıştan Aşağı
12	Üç Dıştan Yukarı	Üç Dıştan Yukarı
13	Üç Dıştan Aşağı	Üç Dıştan Aşağı
14	Üç Yıldız Aşağı	Üç Dıştan Aşağı
15	Üç Dıştan Aşağı	Üç Dıştan Aşağı
16	Üç Dıştan Aşağı	Üç Dıştan Aşağı
17	Üç Dıştan Aşağı	Üç Dıştan Aşağı
18	Üç Dıştan Aşağı	Üç Dıştan Aşağı
19	Akşam Yıldızı	Akşam Yıldızı

ISYAT	Model Sonucu	Gerçek Değer
1	Üç Dıştan Yukarı	Üç İçeride Yukarı
2	Üç Dıştan Yukarı	Üç İçeride Yukarı
3	Üç İçeride Yukarı	Üç İçeride Yukarı
4	Üç İçeride Yukarı	Üç İçeride Yukarı
5	Üç Dıştan Yukarı	Üç Dıştan Yukarı
6	Üç Dıştan Aşağı	Üç Dıştan Aşağı
7	Üç Dıştan Aşağı	Üç Dıştan Aşağı
8	Üç Dıştan Aşağı	Üç Dıştan Aşağı
9	Üç Dıştan Aşağı	Üç Dıştan Aşağı

ISYAT	Model Sonucu	Gerçek Değer
10	Üç Dıştan Yukarı	Üç Dıştan Yukarı
11	Üç Dıştan Aşağı	Üç Dıştan Aşağı
12	Üç Dıştan Yukarı	Üç Dıştan Yukarı
13	Üç Dıştan Aşağı	Üç Dıştan Aşağı
14	Üç Dıştan Aşağı	Üç Dıştan Aşağı
15	Sabah Yıldızı	Sabah Yıldızı
16	Sabah Yıldızı	Sabah Yıldızı

KCHOL	Model Sonucu	Gerçek Değer
1	Üç İçeride Yukarı	Üç İçeride Yukarı
2	Üç Dıştan Yukarı	Üç İçeride Yukarı
3	Üç İçeride Yukarı	Üç İçeride Yukarı
4	Üç İçeride Yukarı	Üç İçeride Yukarı
5	Üç İçeride Aşağı	Üç İçeride Aşağı
6	Üç İçeride Yukarı	Üç İçeride Yukarı
7	Üç Dıştan Aşağı	Üç Dıştan Aşağı
8	Üç Dıştan Aşağı	Üç Dıştan Aşağı
9	Üç Dıştan Aşağı	Üç Dıştan Aşağı
10	Üç Dıştan Aşağı	Üç Dıştan Aşağı
11	Üç Dıştan Aşağı	Üç Dıştan Aşağı
12	Üç Dıştan Yukarı	Üç Dıştan Yukarı
13	Üç Dıştan Aşağı	Üç Dıştan Aşağı
14	Üç Dıştan Aşağı	Üç Dıştan Aşağı
15	Üç Dıştan Aşağı	Üç Dıştan Aşağı
16	Üç Dıştan Yukarı	Üç Dıştan Yukarı
17	Üç Dıştan Aşağı	Üç Dıştan Aşağı
18	Üç Dıştan Aşağı	Üç Dıştan Aşağı
19	Üç Dıştan Aşağı	Üç Dıştan Aşağı
20	Akşam Yıldızı	Akşam Yıldızı
21	Akşam Yıldızı	Akşam Yıldızı

KENT	Model Sonucu	Gerçek Değer
1	Üç İçeride Yukarı	Üç İçeride Yukarı
2	Üç İçeride Aşağı	Üç İçeride Aşağı
3	Üç Dıştan Yukarı	Üç Dıştan Yukarı
4	Üç Dıştan Aşağı	Üç Dıştan Aşağı
5	Üç Dıştan Aşağı	Üç Dıştan Aşağı
6	Üç Dıştan Aşağı	Üç Dıştan Aşağı
7	Üç Dıştan Aşağı	Üç Dıştan Aşağı

KENT	Model Sonucu	Gerçek Değer
8	Üç Dıştan Yukarı	Üç Dıştan Yukarı
9	Üç Dıştan Aşağı	Üç Dıştan Aşağı
10	Üç Dıştan Aşağı	Üç Dıştan Aşağı
11	Üç Dıştan Yukarı	Üç Dıştan Yukarı
12	Üç Dıştan Aşağı	Üç Dıştan Aşağı
13	Akşam Yıldızı	Akşam Yıldızı
14	Akşam Yıldızı	Akşam Yıldızı
15	Sabah Yıldızı	Sabah Yıldızı

KLİNMA	Model Sonucu	Gerçek Değer
1	Üç İçeride Aşağı	Üç İçeride Aşağı
2	Üç Dıştan Aşağı	Üç Dıştan Aşağı
3	Üç Dıştan Aşağı	Üç Dıştan Aşağı
4	Üç Dıştan Aşağı	Üç Dıştan Aşağı
5	Üç Dıştan Aşağı	Üç Dıştan Aşağı
6	Üç Dıştan Aşağı	Üç Dıştan Aşağı
7	Üç Dıştan Yukarı	Üç Dıştan Yukarı
8	Üç Dıştan Aşağı	Üç Dıştan Aşağı
9	Üç Dıştan Yukarı	Üç Dıştan Yukarı
10	Üç İçeride Yukarı	Sabah Yıldızı
11	Üç Yıldız Yukarı	Üç Yıldız Aşağı

KOZAA	Model Sonucu	Gerçek Değer
1	Üç İçeride Yukarı	Üç İçeride Yukarı
2	Üç İçeride Yukarı	Üç İçeride Yukarı
3	Üç Dıştan Aşağı	Üç Dıştan Aşağı
4	Üç Dıştan Aşağı	Üç Dıştan Aşağı
5	Üç Dıştan Aşağı	Üç Dıştan Aşağı
6	Üç Dıştan Aşağı	Üç Dıştan Aşağı
7	Üç Dıştan Aşağı	Üç Dıştan Aşağı
8	Üç Dıştan Aşağı	Üç Dıştan Aşağı
9	Üç Dıştan Yukarı	Üç Dıştan Yukarı
10	Akşam Yıldızı	Akşam Yıldızı
11	Akşam Yıldızı	Akşam Yıldızı
12	Akşam Yıldızı	Akşam Yıldızı
13	Akşam Yıldızı	Akşam Yıldızı
14	Akşam Yıldızı	Akşam Yıldızı
15	Üç İçeride Yukarı	Sabah Yıldızı
16	Sabah Yıldızı	Sabah Yıldızı

KOZAL	Model Sonucu	Gerçek Değer
1	Üç İçeride Yukarı	Üç İçeride Yukarı
2	Üç İçeride Yukarı	Üç İçeride Yukarı
3	Üç İçeride Yukarı	Üç İçeride Yukarı
4	Üç İçeride Yukarı	Üç İçeride Yukarı
5	Üç İçeride Yukarı	Üç İçeride Yukarı
6	Üç İçeride Yukarı	Üç İçeride Yukarı
7	Üç İçeride Yukarı	Üç İçeride Yukarı
8	Üç Dıştan Aşağı	Üç Dıştan Aşağı
9	Üç Dıştan Aşağı	Üç Dıştan Aşağı
10	Üç Dıştan Aşağı	Üç Dıştan Aşağı
11	Üç Dıştan Aşağı	Üç Dıştan Aşağı
12	Üç Dıştan Aşağı	Üç Dıştan Aşağı
13	Üç Dıştan Aşağı	Üç Dıştan Aşağı
14	Üç Dıştan Aşağı	Üç Dıştan Aşağı
15	Akşam Yıldızı	Akşam Yıldızı
16	Akşam Yıldızı	Akşam Yıldızı
17	Akşam Yıldızı	Akşam Yıldızı
18	Akşam Yıldızı	Akşam Yıldızı
19	Üç İçeride Yukarı	Sabah Yıldızı

KRDMD	Model Sonucu	Gerçek Değer
1	Üç İçeride Aşağı	Üç İçeride Aşağı
2	Üç İçeride Yukarı	Üç İçeride Yukarı
3	Üç Dıştan Yukarı	Üç Dıştan Yukarı
4	Üç Dıştan Aşağı	Üç Dıştan Aşağı
5	Üç Dıştan Aşağı	Üç Dıştan Aşağı
6	Üç Dıştan Aşağı	Üç Dıştan Aşağı
7	Üç Dıştan Aşağı	Üç Dıştan Aşağı
8	Üç Dıştan Aşağı	Üç Dıştan Aşağı
9	Üç Dıştan Aşağı	Üç Dıştan Aşağı
10	Üç Dıştan Aşağı	Üç Dıştan Aşağı
11	Üç Dıştan Aşağı	Üç Dıştan Aşağı
12	Üç Dıştan Aşağı	Üç Dıştan Aşağı
13	Üç Dıştan Aşağı	Üç Dıştan Aşağı
14	Üç Dıştan Yukarı	Üç Dıştan Yukarı
15	Üç Dıştan Aşağı	Üç Dıştan Aşağı
16	Akşam Yıldızı	Akşam Yıldızı

PETKM	Model Sonucu	Gerçek Değer
1	Üç İçeride Yukarı	Üç İçeride Yukarı
2	Üç İçeride Aşağı	Üç İçeride Aşağı
3	Üç İçeride Aşağı	Üç İçeride Aşağı
4	Üç Dıştan Yukarı	Üç İçeride Yukarı
5	Üç İçeride Yukarı	Üç İçeride Yukarı
6	Üç Dıştan Yukarı	Üç Dıştan Yukarı
7	Üç Dıştan Aşağı	Üç Dıştan Aşağı
8	Üç Dıştan Aşağı	Üç Dıştan Aşağı
9	Üç Dıştan Aşağı	Üç Dıştan Aşağı
10	Üç Dıştan Aşağı	Üç Dıştan Aşağı
11	Üç Dıştan Aşağı	Üç Dıştan Aşağı
12	Akşam Yıldızı	Akşam Yıldızı
13	Sabah Yıldızı	Sabah Yıldızı
14	Sabah Yıldızı	Sabah Yıldızı

PGSUS	Model Sonucu	Gerçek Değer
1	Üç İçeride Yukarı	Üç İçeride Yukarı
2	Üç İçeride Yukarı	Üç İçeride Yukarı
3	Üç İçeride Yukarı	Üç İçeride Yukarı
4	Üç İçeride Aşağı	Üç İçeride Aşağı
5	Üç Dıştan Yukarı	Üç İçeride Yukarı
6	Üç İçeride Aşağı	Üç İçeride Aşağı
7	Üç Dıştan Aşağı	Üç Dıştan Aşağı
8	Üç Dıştan Aşağı	Üç Dıştan Aşağı
9	Üç Dıştan Yukarı	Üç Dıştan Yukarı
10	Üç Dıştan Yukarı	Üç Dıştan Yukarı
11	Akşam Yıldızı	Akşam Yıldızı

QNBFB	Model Sonucu	Gerçek Değer
1	Üç İçeride Aşağı	Üç İçeride Aşağı
2	Üç Dıştan Yukarı	Üç İçeride Yukarı
3	Üç Dıştan Yukarı	Üç Dıştan Yukarı
4	Üç Dıştan Yukarı	Üç Dıştan Yukarı
5	Üç Dıştan Aşağı	Üç Dıştan Aşağı
6	Üç Dıştan Yukarı	Üç Dıştan Yukarı
7	Akşam Yıldızı	Akşam Yıldızı
8	Akşam Yıldızı	Akşam Yıldızı
9	Akşam Yıldızı	Akşam Yıldızı
10	Akşam Yıldızı	Akşam Yıldızı

QNBFB	Model Sonucu	Gerçek Değer
11	Sabah Yıldızı	Sabah Yıldızı

QNBFL	Model Sonucu	Gerçek Değer
1	Üç İçeride Yukarı	Üç İçeride Yukarı
2	Üç İçeride Aşağı	Üç İçeride Aşağı
3	Üç İçeride Aşağı	Üç İçeride Aşağı
4	Üç İçeride Yukarı	Üç İçeride Yukarı
5	Üç Dıştan Aşağı	Üç Dıştan Aşağı
6	Üç Dıştan Aşağı	Üç Dıştan Aşağı
7	Üç Dıştan Aşağı	Üç Dıştan Aşağı
8	Üç Dıştan Yukarı	Üç Dıştan Yukarı
9	Üç Dıştan Aşağı	Üç Dıştan Aşağı
10	Akşam Yıldızı	Akşam Yıldızı
11	Üç Yıldız Yukarı	Üç Yıldız Yukarı

SAHOL	Model Sonucu	Gerçek Değer
1	Üç İçeride Aşağı	Üç İçeride Aşağı
2	Üç İçeride Aşağı	Üç İçeride Aşağı
3	Üç İçeride Aşağı	Üç İçeride Aşağı
4	Üç İçeride Aşağı	Üç İçeride Aşağı
5	Üç İçeride Yukarı	Üç İçeride Yukarı
6	Üç Dıştan Yukarı	Üç Dıştan Yukarı
7	Üç Dıştan Aşağı	Üç Dıştan Aşağı
8	Üç Dıştan Yukarı	Üç Dıştan Yukarı
9	Üç Dıştan Aşağı	Üç Dıştan Aşağı
10	Üç Dıştan Aşağı	Üç Dıştan Aşağı
11	Üç Dıştan Yukarı	Üç Dıştan Yukarı
12	Üç Dıştan Aşağı	Üç Dıştan Aşağı
13	Üç Dıştan Yukarı	Üç Dıştan Yukarı
14	Üç Dıştan Aşağı	Üç Dıştan Aşağı
15	Üç Dıştan Aşağı	Üç Dıştan Aşağı
16	Üç Dıştan Aşağı	Üç Dıştan Aşağı
17	Üç Dıştan Aşağı	Üç Dıştan Aşağı
18	Üç Dıştan Aşağı	Üç Dıştan Aşağı
19	Üç Dıştan Aşağı	Üç Dıştan Aşağı
20	Üç Dıştan Aşağı	Üç Dıştan Aşağı
21	Sabah Yıldızı	Sabah Yıldızı
22	Sabah Yıldızı	Sabah Yıldızı
23	Sabah Yıldızı	Sabah Yıldızı

SASA	Model Sonucu	Gerçek Değer
1	Üç İçeride Aşağı	Üç İçeride Aşağı
2	Üç İçeride Aşağı	Üç İçeride Aşağı
3	Üç İçeride Yukarı	Üç İçeride Yukarı
4	Üç İçeride Yukarı	Üç İçeride Yukarı
5	Üç İçeride Yukarı	Üç İçeride Yukarı
6	Üç Dıştan Aşağı	Üç Dıştan Aşağı
7	Üç Dıştan Yukarı	Üç Dıştan Yukarı
8	Üç Dıştan Yukarı	Üç Dıştan Yukarı
9	Üç Dıştan Aşağı	Üç Dıştan Aşağı
10	Üç Dıştan Yukarı	Üç Dıştan Yukarı
11	Üç Dıştan Aşağı	Üç Dıştan Aşağı
12	Üç Dıştan Aşağı	Üç Dıştan Aşağı
13	Üç Dıştan Aşağı	Üç Dıştan Aşağı
14	Üç Dıştan Aşağı	Üç Dıştan Aşağı
15	Üç İçeride Aşağı	Akşam Yıldızı

SISE	Model Sonucu	Gerçek Değer
1	Üç İçeride Yukarı	Üç İçeride Yukarı
2	Üç İçeride Aşağı	Üç İçeride Aşağı
3	Üç İçeride Aşağı	Üç İçeride Aşağı
4	Üç Dıştan Yukarı	Üç Dıştan Yukarı
5	Üç Dıştan Aşağı	Üç Dıştan Aşağı
6	Üç Dıştan Aşağı	Üç Dıştan Aşağı
7	Üç Dıştan Yukarı	Üç Dıştan Yukarı
8	Üç Dıştan Aşağı	Üç Dıştan Aşağı
9	Üç Dıştan Aşağı	Üç Dıştan Aşağı
10	Üç Dıştan Yukarı	Üç Dıştan Yukarı
11	Üç Dıştan Aşağı	Üç Dıştan Aşağı
12	Üç Dıştan Aşağı	Üç Dıştan Aşağı
13	Üç Dıştan Yukarı	Üç Dıştan Yukarı
14	Akşam Yıldızı	Akşam Yıldızı
15	Akşam Yıldızı	Akşam Yıldızı
18	Akşam Yıldızı	Akşam Yıldızı
19	Sabah Yıldızı	Sabah Yıldızı
20	Sabah Yıldızı	Sabah Yıldızı

SNKRN	Model Sonucu	Gerçek Değer
1	Üç İçeride Yukarı	Üç İçeride Yukarı
2	Üç İçeride Aşağı	Üç İçeride Aşağı

SNKRN	Model Sonucu	Gerçek Değer
3	Üç Dıştan Yukarı	Üç Dıştan Yukarı
4	Üç Dıştan Aşağı	Üç Dıştan Aşağı
5	Üç Dıştan Yukarı	Üç Dıştan Yukarı
6	Üç Dıştan Aşağı	Üç Dıştan Aşağı
7	Üç Dıştan Aşağı	Üç Dıştan Aşağı
8	Üç Dıştan Yukarı	Üç Dıştan Yukarı
9	Üç Dıştan Yukarı	Üç Dıştan Yukarı
10	Üç Dıştan Yukarı	Üç Dıştan Yukarı
11	Sabah Yıldızı	Sabah Yıldızı
12	Sabah Yıldızı	Sabah Yıldızı

TAVHL	Model Sonucu	Gerçek Değer
1	Üç İçeride Yukarı	Üç İçeride Yukarı
2	Üç İçeride Yukarı	Üç İçeride Yukarı
3	Üç Dıştan Aşağı	Üç Dıştan Aşağı
4	Üç Dıştan Aşağı	Üç Dıştan Aşağı
5	Üç Dıştan Aşağı	Üç Dıştan Aşağı
6	Üç Dıştan Aşağı	Üç Dıştan Aşağı
7	Üç Dıştan Aşağı	Üç Dıştan Aşağı
8	Üç Dıştan Aşağı	Üç Dıştan Aşağı
9	Üç Dıştan Aşağı	Üç Dıştan Aşağı
10	Üç Dıştan Aşağı	Üç Dıştan Aşağı
11	Üç Dıştan Aşağı	Üç Dıştan Aşağı
12	Üç Dıştan Aşağı	Üç Dıştan Aşağı
13	Üç Dıştan Aşağı	Üç Dıştan Aşağı
14	Üç Dıştan Aşağı	Üç Dıştan Aşağı
15	Akşam Yıldızı	Akşam Yıldızı
16	Akşam Yıldızı	Akşam Yıldızı

TBORG	Model Sonucu	Gerçek Değer
1	Üç İçeride Yukarı	Üç İçeride Yukarı
2	Üç Dıştan Aşağı	Üç Dıştan Aşağı
3	Üç Dıştan Aşağı	Üç Dıştan Aşağı
4	Üç Dıştan Aşağı	Üç Dıştan Aşağı
5	Üç Dıştan Aşağı	Üç Dıştan Aşağı
6	Üç Dıştan Aşağı	Üç Dıştan Aşağı
7	Akşam Yıldızı	Akşam Yıldızı
8	Sabah Yıldızı	Sabah Yıldızı

TCELL	Model Sonucu	Gerçek Değer
1	Üç İçeride Yukarı	Üç İçeride Yukarı
2	Üç İçeride Yukarı	Üç İçeride Yukarı
3	Üç İçeride Yukarı	Üç İçeride Yukarı
4	Üç Dıştan Aşağı	Üç Dıştan Aşağı
5	Üç Dıştan Aşağı	Üç Dıştan Aşağı
6	Üç Dıştan Aşağı	Üç Dıştan Aşağı
7	Üç Dıştan Aşağı	Üç Dıştan Aşağı
8	Üç Dıştan Aşağı	Üç Dıştan Aşağı
9	Üç Dıştan Aşağı	Üç Dıştan Aşağı
10	Sabah Yıldızı	Sabah Yıldızı
11	Sabah Yıldızı	Sabah Yıldızı

THYAO	Model Sonucu	Gerçek Değer
1	Üç İçeride Yukarı	Üç İçeride Yukarı
2	Üç İçeride Aşağı	Üç İçeride Aşağı
3	Üç İçeride Aşağı	Üç İçeride Aşağı
4	Üç İçeride Aşağı	Üç İçeride Aşağı
5	Üç Dıştan Aşağı	Üç Dıştan Aşağı
6	Üç Dıştan Aşağı	Üç Dıştan Aşağı
7	Üç Dıştan Aşağı	Üç Dıştan Aşağı
8	Üç Dıştan Aşağı	Üç Dıştan Aşağı
9	Üç Dıştan Aşağı	Üç Dıştan Aşağı
10	Üç Dıştan Aşağı	Üç Dıştan Aşağı
11	Üç Dıştan Aşağı	Üç Dıştan Aşağı
12	Üç Dıştan Aşağı	Üç Dıştan Aşağı
13	Üç Dıştan Yukarı	Üç Dıştan Yukarı
14	Akşam Yıldızı	Akşam Yıldızı

TKFEN	Model Sonucu	Gerçek Değer
1	Üç İçeride Yukarı	Üç İçeride Yukarı
2	Üç İçeride Yukarı	Üç İçeride Yukarı
3	Üç İçeride Yukarı	Üç İçeride Yukarı
4	Üç İçeride Yukarı	Üç İçeride Yukarı
5	Üç İçeride Yukarı	Üç İçeride Yukarı
6	Üç İçeride Yukarı	Üç İçeride Yukarı
7	Üç İçeride Yukarı	Üç İçeride Yukarı
8	Üç Dıştan Aşağı	Üç Dıştan Aşağı
9	Üç Dıştan Aşağı	Üç Dıştan Aşağı
10	Üç Dıştan Yukarı	Üç Dıştan Yukarı

TKFEN	Model Sonucu	Gerçek Değer
11	Üç Dıştan Aşağı	Üç Dıştan Aşağı
12	Üç Dıştan Aşağı	Üç Dıştan Aşağı
13	Üç Dıştan Aşağı	Üç Dıştan Aşağı
14	Üç Dıştan Yukarı	Üç Dıştan Yukarı
15	Üç Dıştan Aşağı	Üç Dıştan Aşağı
16	Üç Dıştan Aşağı	Üç Dıştan Aşağı
17	Üç Dıştan Aşağı	Üç Dıştan Aşağı
18	Üç Dıştan Yukarı	Üç Dıştan Yukarı
19	Akşam Yıldızı	Akşam Yıldızı
20	Akşam Yıldızı	Akşam Yıldızı
21	Sabah Yıldızı	Sabah Yıldızı
22	Üç İçeride Yukarı	Sabah Yıldızı

TOASO	Model Sonucu	Gerçek Değer
1	Üç İçeride Yukarı	Üç İçeride Yukarı
2	Üç İçeride Yukarı	Üç İçeride Yukarı
3	Üç Dıştan Yukarı	Üç İçeride Yukarı
4	Üç İçeride Yukarı	Üç İçeride Yukarı
5	Üç Dıştan Yukarı	Üç Dıştan Yukarı
6	Üç Dıştan Aşağı	Üç Dıştan Aşağı
7	Üç Dıştan Yukarı	Üç Dıştan Yukarı
8	Üç Dıştan Aşağı	Üç Dıştan Aşağı
9	Üç Dıştan Yukarı	Üç Dıştan Yukarı
10	Üç Dıştan Aşağı	Üç Dıştan Aşağı
11	Üç Dıştan Aşağı	Üç Dıştan Aşağı
12	Üç Dıştan Aşağı	Üç Dıştan Aşağı
13	Akşam Yıldızı	Akşam Yıldızı
14	Sabah Yıldızı	Sabah Yıldızı
15	Üç İçeride Yukarı	Sabah Yıldızı

TTKOM	Model Sonucu	Gerçek Değer
1	Üç Dıştan Yukarı	Üç İçeride Yukarı
2	Üç İçeride Yukarı	Üç İçeride Yukarı
3	Üç İçeride Yukarı	Üç İçeride Yukarı
4	Üç İçeride Yukarı	Üç İçeride Yukarı
5	Üç Dıştan Yukarı	Üç İçeride Yukarı
6	Üç Dıştan Yukarı	Üç İçeride Yukarı
7	Üç İçeride Yukarı	Üç İçeride Yukarı
8	Üç Dıştan Aşağı	Üç Dıştan Aşağı

TTKOM	Model Sonucu	Gerçek Değer
9	Üç Dıştan Aşağı	Üç Dıştan Aşağı
10	Üç Dıştan Aşağı	Üç Dıştan Aşağı
11	Üç Dıştan Aşağı	Üç Dıştan Aşağı
12	Üç Dıştan Yukarı	Üç Dıştan Yukarı
13	Üç Dıştan Aşağı	Üç Dıştan Aşağı
14	Üç Dıştan Aşağı	Üç Dıştan Aşağı
15	Üç Dıştan Aşağı	Üç Dıştan Aşağı
16	Üç Dıştan Aşağı	Üç Dıştan Aşağı

TUPRS	Model Sonucu	Gerçek Değer
1	Üç İçeride Yukarı	Üç İçeride Yukarı
2	Üç İçeride Yukarı	Üç İçeride Yukarı
3	Üç İçeride Aşağı	Üç İçeride Aşağı
4	Üç Dıştan Yukarı	Üç Dıştan Yukarı
5	Üç Dıştan Aşağı	Üç Dıştan Aşağı
6	Üç Dıştan Aşağı	Üç Dıştan Aşağı
7	Üç Dıştan Aşağı	Üç Dıştan Aşağı
8	Üç Dıştan Aşağı	Üç Dıştan Aşağı
9	Üç Dıştan Aşağı	Üç Dıştan Aşağı
10	Üç Dıştan Aşağı	Üç Dıştan Aşağı
11	Üç Dıştan Aşağı	Üç Dıştan Aşağı
12	Üç Dıştan Aşağı	Üç Dıştan Aşağı
13	Üç Dıştan Aşağı	Üç Dıştan Aşağı
14	Üç Dıştan Aşağı	Üç Dıştan Aşağı
15	Üç Dıştan Yukarı	Üç Dıştan Yukarı
16	Akşam Yıldızı	Akşam Yıldızı
17	Akşam Yıldızı	Akşam Yıldızı
18	Akşam Yıldızı	Akşam Yıldızı
19	Akşam Yıldızı	Akşam Yıldızı
20	Akşam Yıldızı	Akşam Yıldızı
21	Akşam Yıldızı	Akşam Yıldızı

VESTL	Model Sonucu	Gerçek Değer
1	Üç İçeride Yukarı	Üç İçeride Yukarı
2	Üç İçeride Yukarı	Üç İçeride Yukarı
3	Üç İçeride Yukarı	Üç İçeride Yukarı
4	Üç İçeride Yukarı	Üç İçeride Yukarı
5	Üç Dıştan Aşağı	Üç Dıştan Aşağı
6	Üç Dıştan Aşağı	Üç Dıştan Aşağı

VESTL	Model Sonucu	Gerçek Değer
7	Üç Dıştan Aşağı	Üç Dıştan Aşağı
8	Üç Dıştan Aşağı	Üç Dıştan Aşağı
9	Üç Dıştan Yukarı	Üç Dıştan Yukarı
10	Üç Dıştan Aşağı	Üç Dıştan Aşağı
11	Üç Dıştan Aşağı	Üç Dıştan Aşağı
12	Üç Dıştan Aşağı	Üç Dıştan Aşağı
13	Üç İçeride Aşağı	Akşam Yıldızı
14	Sabah Yıldızı	Sabah Yıldızı

YKBNK	Model Sonucu	Gerçek Değer
1	Üç İçeride Yukarı	Üç İçeride Yukarı
2	Üç İçeride Yukarı	Üç İçeride Yukarı
3	Üç İçeride Aşağı	Üç İçeride Aşağı
4	Üç İçeride Aşağı	Üç İçeride Aşağı
5	Üç İçeride Aşağı	Üç İçeride Aşağı
6	Üç İçeride Aşağı	Üç İçeride Aşağı
7	Üç Dıştan Aşağı	Üç Dıştan Aşağı
8	Üç Dıştan Aşağı	Üç Dıştan Aşağı
9	Üç Dıştan Aşağı	Üç Dıştan Aşağı
10	Üç Dıştan Aşağı	Üç Dıştan Aşağı
11	Üç Dıştan Aşağı	Üç Dıştan Aşağı
12	Üç Dıştan Yukarı	Üç Dıştan Yukarı
13	Üç Dıştan Yukarı	Üç Dıştan Yukarı
14	Üç Dıştan Aşağı	Üç Dıştan Aşağı
15	Üç Dıştan Aşağı	Üç Dıştan Aşağı
16	Üç Dıştan Aşağı	Üç Dıştan Aşağı
17	Üç Dıştan Aşağı	Üç Dıştan Aşağı
18	Üç Dıştan Aşağı	Üç Dıştan Aşağı
19	Akşam Yıldızı	Akşam Yıldızı
20	Akşam Yıldızı	Akşam Yıldızı
21	Sabah Yıldızı	Sabah Yıldızı
22	Sabah Yıldızı	Sabah Yıldızı
23	Sabah Yıldızı	Sabah Yıldızı

ZEDUR	Model Sonucu	Gerçek Değer
1	Üç İçeride Yukarı	Üç İçeride Yukarı
2	Üç İçeride Yukarı	Üç İçeride Yukarı
3	Üç İçeride Aşağı	Üç İçeride Aşağı
4	Üç İçeride Yukarı	Üç İçeride Yukarı

5	Üç Dıştan Aşağı	Üç Dıştan Aşağı
6	Üç Dıştan Aşağı	Üç Dıştan Aşağı
7	Üç Dıştan Yukarı	Üç Dıştan Yukarı
8	Üç Dıştan Aşağı	Üç Dıştan Aşağı
9	Üç Dıştan Yukarı	Üç Dıştan Yukarı
10	Üç Dıştan Aşağı	Üç Dıştan Aşağı
11	Akşam Yıldızı	Akşam Yıldızı
12	Sabah Yıldızı	Sabah Yıldızı



KAYNAKÇA

- Achelis, S. B. (2001). *Technical Analysis From A to Z*. Second Edition. New York: McGraw Hill.
- Ackley, D. H., G. E. Hinton ve T. J. Sejnowski. (1985). A Learning Algorithm for Boltzmann Machines. *Cognitive Science*. 9, 147-169.
- Aggarwal, C. C. (2018). *Neural Networks and Deep Learning*. New York: Springer. <https://doi.org/10.1007/978-3-319-94463-0>
- Alammar, J. (2016). A Visual and Interactive Guide to the Basics of Neural Networks. <https://jalammar.github.io/visual-interactive-guide-basics-neural-networks/> (21 Haziran 2021)
- Albon, C. (2018). *Machine Learning with Python Cookbook*. ABD: O'Reilly Media, Inc.
- Aliyev, V. (2020). A Definitive Explanation to the Hinge Loss for Support Vector Machines. <https://towardsdatascience.com/a-definitive-explanation-to-hinge-loss-for-support-vector-machines-ab6d8d3178f1> (18 Mart 2021).
- Alom, M. Z., T. M. Taha, C. Yakopcic, S. Westberg, P. Sidike, M. S. Nasrin ve V. K. Asari. (2019). A State-of-the-art Survey on Deep Learning Theory and Architectures. *Electronics*. 8.3, 292. <https://doi.org/10.3390/electronics8030292>
- Andriyanto, A., A. Wibowo ve N. Z. Abidin. (2020). Sectoral Stock Prediction Using Convolutional Neural Networks With Candlestick Patterns as Input Images. *International Journal*. 8.6. <https://doi.org/10.47065/josyc.v3i4.2133>
- Arif, T. M. (2020). Introduction to Deep Learning for Engineers: Using Python and Google Cloud Platform. *Synthesis Lectures on Mechanical Engineering*. 5.3, 1-109.
- Armoogum, S., ve X. Li. (2019). "Big Data Analytics and Deep Learning in Bioinformatics With Hadoop". In *Deep Learning and Parallel Computing Environment for Bioengineering Systems* (s. 17-36). Academic Press.
- Avcılar, M. Y., ve E. Yakut. (2015). Yapay Sinir Ağları Çoklu Lojistik Regresyon ve Çoklu Diskriminant Analiz Yöntemlerinden Yararlanarak Yerel Seçimlerde Seçmen Tercihlerinin Belirlenmesi: Osmaniye ili Uygulaması. *Alanya İslam Fakültesi Dergisi*. 7.2, 207-224.
- Bayraktar, A. (2012). Etkin Piyasalar Hipotezi. *Aksaray Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*. 4.1, 37-47.
- Bengio, Y. (2009). Learning Deep Architectures for AI. *Found. Trends Mach. Learning*. 2, 1-127.

- Bengio, Y., A. Courville ve P. Vincent. (2013). Representation Learning: A Review and New Perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.* 35, 1798-1828.
- Bengio, Y., P. Simard ve P. Frasconi. (1994). Learning Long-term Dependencies With Gradient Descent is Difficult. *IEEE Transactions on Neural Networks.* 5.2, 157-166.
- Bengio, Y., Y. LeCun ve G. Hinton. (2015). Deep Learning. *Nature.* 521, 436-444.
- Bezdan, T., ve N. Baćanin Džakula. (2019). Convolutional Neural Network Layers and Architectures. *Paper presented at Sinteza 2019 - International Scientific Conference on Information Technology and Data Related Research* (s. 445-451). <http://doi.org/10.15308/Sinteza-2019-445-451>
- Biroğul, S., G. Temür ve U. Kose. (2020). YOLO Object Recognition Algorithm and “Buy-Sell Decision” Model Over 2D Candlestick Charts. *IEEE Access.* 8, 91894-91915.
- Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Oxford: Clarendon Press.
- Bjorck, N., C. P., Gomes, B. Selman ve K. Q. Weinberger. (2018). Understanding Batch Normalization. *Advances in Neural Information Processing Systems*, 31, Montréal, Canada.
- Brownlee, J. (2019). Loss and Loss Functions for Training Deep Learning Neural Networks. <https://machinelearningmastery.com/loss-and-loss-functions-for-training-deep-learning-neural-networks/> (18 Mart 2021).
- Bulkowski, T. N. (2008). *Encyclopedia of Candlestick Charts*. John Wiley & Sons.
- Calin, O. (2020). *Deep Learning Architectures*. Springer International Publishing.
- Chen, J. H., C. H. Wu, Y. C. Tsai ve S. Y. C. Chen. (2022, January). “Explainable Digital Currency Candlestick Pattern AI Learner”. In *2022 14th International Conference on Knowledge and Smart Technology (KST)* (s. 91-96). IEEE.
- Chen, J. H., s. Y. C. Chen, Y. C. Tsai ve C. S. Shur. (2020). Explainable Deep Convolutional Candlestick Learner. *arXiv preprint arXiv:2001.02767*.
- Chen, J. H., ve Y. C. Tsai. (2020). Encoding Candlesticks as Images for Pattern Classification Using Convolutional Neural Networks. *Financial Innovation.* 6.1, 1-19.
- Chen, J. H., ve Y. C. Tsai. (2022). Dynamic Deep Convolutional Candlestick Learner. *arXiv preprint arXiv:2201.08669*.
- Chen, X.-W., ve X. Lin. (2014). Big Data Deep Learning: Challenges and Perspectives. *IEEE Access.* 2, 514-525.

- Cho, K., B. Van Merriënboer, D. Bahdanau ve Y. Bengio. (2014). On the Properties of Neural Machine Translation: Encoder-decoder Approaches. *arXiv preprint arXiv:1409.1259*.
- Chollet, F. (2018). *Deep Learning With Python*. New York: Manning Publications Co.
- Chung, J., C. Gulcehre, K. Cho ve Y. Bengio. (2014). Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *arXiv preprint arXiv:1412.3555*.
- Clevert, D. A., T. Unterthiner ve S. Hochreiter. (2016). Fast and Accurate Deep Network Learning by Exponential Linear Units (Elus). *arXiv preprint arXiv:1511.07289*.
- Cruse, H. (2006). *Neural Networks as Cybernetic Systems*. Bielefeld: Brains, Minds & Media.
- Çelik, T. T., ve O. Taş. (2007). Etkin Piyasa Hipotezi ve Gelişmekte Olan Hisse Senedi Piyasaları. *İTÜ Dergisi*. 4.2, 11-22.
- Çetinyokuş, T., ve H. Gökçen (2013). Borsada Göstergelerle Teknik Analiz İçin Bir Karar Destek Sistemi. *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*, 17(1), 43-58.
- Da Silva, I. N., D. Hernane Spatti, R. Andrade Flauzino, L. H. B. Liboni, S. F. dos Reis Alves, I. N. da Silva ve S. F. dos Reis Alves. (2017). *Artificial Neural Network Architectures and Training Processes* (s. 21-28). Springer International Publishing.
- Dabbura, I. (2017). Gradient Descent Algorithm and its Variants. <https://towardsdatascience.com/gradient-descent-algorithm-and-its-variants-10f652806a3> (07 Eylül 2021).
- Dayhoff, J. E., ve J. M. DeLeo. (2001). Artificial Neural Networks: Opening the Black Box. *Conference on Prognostic Factors and Staging in Cancer Management: Contributions of Artificial Neural Networks and Other Statistical Methods*, Vol.91, No.S8. [https://doi.org/10.1002/1097-0142\(20010415\)91:8+<1615::AID-CNCR1175>3.0.CO;2-L](https://doi.org/10.1002/1097-0142(20010415)91:8+<1615::AID-CNCR1175>3.0.CO;2-L)
- Di Lorenzo, R. (2013). *Basic Technical Analysis of Financial Markets*. Milan, Italy: Springer.
- Elman, J. (1990). Finding Structure in Time. *Cognitive Science*. 14, 179-211.
- Erdinç, Y. (2004). *Teknik Analiz ve Yatırımcı Sorgulanıyor*. Ankara: Siyasal Kitabevi.
- Fama, E. (1965). The Behavior of Stock-Market Prices. *The JOURNAL of Business*. 38.1, 34-105.

- Fama, E. (1970). Efficient Capital Markets: A Review of Theory and Empirical Work. *The Journal of Finance*. 25.2, 383-417.
- Fausett, L. V. (1993). *Fundamentals of Neural Networks: Architectures, Algorithms and Applications*. Pearson Education India.
- Feng, W., N. Guan, Y. Li, X. Zhang ve Z. Luo. (2017, May). "Audio Visual Speech Recognition With Multimodal Recurrent Neural Networks". In *2017 International Joint Conference On Neural Networks (IJCNN)* (s. 681-688). IEEE.
- Freund, Y., ve R. E. Schapire. (1995). A Desicion-Theoretic Generalization of On-Line Learning and an Application to Boosting. In: *European Conference on Computational Learning Theory* (s. 23-37). Springer, Berlin, Heidelberg.
- Fukushima, K. (1980). Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position. *Biological Cybernetics*. 36, 193-202.
- Fukushima, K. (1988). Neocognitron: A Hierarchical Neural Network Capable of Visual Pattern Recognition. *Neural Netw.* 1, 119-130.
- Gabor, D. (1946). "Theory of Communication. Part 1: The Analysis of Information". *Journal of the Institution of Electrical Engineers-Part III: radio and Communication Engineering*, 93.26, 429-441.
- Garbin, C., X. Zhu ve O. Marques. (2020). Dropout vs. Batch Normalization: An Empirical Study of Their Impact to Deep Learning. *Multimedia Tools and Applications*. 79.19, 12777-12815.
- Geron, A. (2017). "Hands-on Machine Learning With Scikit-Learn and Tensorflow: Concepts". *Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, Inc.
- Ghandi, A. (2021). Data Augmentation | How to use Deep Learning When You Have Limited Data-Part 2. <https://nanonets.com/blog/data-augmentation-how-to-use-deep-learning-when-you-have-> (11 Temmuz 2021).
- Goodfellow, I., Y. Bengio, A. Courville ve Y. Bengio. (2016). *Deep Learning*. Vol.1, No.2. Cambridge: MIT Press.
- Graves, A. (2012). "Supervised Sequence Labelling". In *Supervised Sequence Labelling With Recurrent Neural Networks* (s. 5-13). Berlin, Heidelberg: Springer.
- Grover, P. (2018). Regression Loss Functions All Machine Learners Should Know. <https://heartbeat.fritz.ai/5-regression-loss-functions-all-machine-learners-should-know-4fb140e9d4b0> (11 Mayıs 2021).

- Guo, S. J., F. C. Hsu ve C. C. Hung. (2018, December). “Deep Candlestick Predictor: A Framework Toward Forecasting The Price Movement From Candlestick Charts”. In *2018 9th International Symposium on Parallel Architectures, Algorithms and Programming (PAAP)* (s. 219-226). IEEE.
- Gürsakal, N. (2017). *Makine Öğrenmesi ve Derin Öğrenme*. Bursa: Dora Yayıncılık.
- Gürsakal, N. (2018). *Makine Öğrenmesi*. Bursa: Dora Yayıncılık.
- Hastie, T., R. Tibshirani ve J. Friedman. (2008). “The Elements of Statistical Learning Data Mining”. *Inference and Prediction*. Canada: Springer.
- Haykin, S. (2009). *Neural Networks and Learning Machines* (3. Ed.). New Jersey: Pearson Education.
- He, K., X. Zhang, S. Ren ve J. Sun. (2016). Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (s. 770-778). IEEE. <http://doi.org/10.1109/CVPR.2016.90>
- Heaton, J. (2015). *Artificial Intelligence for Humans: Vol. 3: Neural Networks and Deep Learning*. Chesterfield: Heaton Research Inc.
- Hebb, D. (1949). *The Organization of Behavior*. New York: John Wiley.
- Hinton, G. E., N. Srivastava, A. Krizhevsky, I. Sutskever ve R. R. Salakhutdinov. (2012). Improving Neural Networks by Preventing Co-adaptation of Feature Detectors. *arXiv preprint arXiv:1207.0580*.
- Ho, T. T., ve Y. Huang. (2021). Stock Price Movement Prediction Using Sentiment Analysis and CandleStick Chart Representation. *Sensors*. 21.23, 7957.
- Hochreiter, S. (1998). The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*. 6.02, 107-116.
- Hochreiter, S., ve J. Schmidhuber. (1997). Long Short-Term Memory. *Neural Computation*, 9.8, 1735-1780.
- Hopfield, J. J. (1982). Neural Networks and Physical Systems With Emergent Collective Computational Abilities. *Proceedings of the National Academy of Sciences*. 79.8, 2554-2558.
- Hu, G., Y. Hu, K. Yang, Z. Yu, F. Sung, Z. Zhang ve Q. Miemie. (2018, April). Deep Stock Representation Learning: From Candlestick Charts to Investment Decisions. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (s. 2706-2710). IEEE.
- Hubel, D. H., ve T. N. Wiesel,. (1959). Receptive Fields of Single Neurones in THE Cat’s Striate Cortex. *The Journal of Physiology*. 148.3, 574.

- Hubel, D. H., ve T. N. Wiesel. (1962). Receptive Fields, Binocular Interaction and Functional Architecture in a Cat's Visual Cortex. *Journal of Physiology (London)*. 160, 106-154.
- Hung, C. C., ve Y. J. Chen. (2021). DPP: Deep Predictor for Price Movement From Candlestick Charts. *Plos One*. 16.6, e0252404.
- Hung, C. C., Y. J. Chen, S. J. Guo ve F. C. Hsu. (2020). Predicting the Price Movement from Candlestick Charts: A CNN-Based Approach. *International Journal of Ad Hoc and Ubiquitous Computing*. 34.2, 111-120.
- Ibrahim, Z., D. Isa, Z. Idrus, Z. Kasiran ve R. Roslan. (2019, August). "Evaluation of Pooling Layers in Convolutional Neural Network for Script Recognition". In *International Conference on Soft Computing in Data Science* (s. 121-129). Singapore: Springer.
- Imamura, A., ve N. Arizumi. (2021, December). Gabor Filter Incorporated CNN for Compression. In *2021 36th International Conference on Image and Vision Computing New Zealand (IVCNZ)* (s. 1-5). IEEE.
- Ioffe, S., ve C. Szegedy. (2015, June). "Batch Normalization: Accelerating deep network Training by Reducing Internal Covariate Shift". In *International Conference on Machine Learning* (s. 448-456). PMLR.
- İnik, Ö., ve F. Ülker. (2017). Derin Öğrenme ve Görüntü Analizinde Kullanılan Derin Öğrenme Modelleri. *Gaziosmanpaşa Bilimsel Araştırma Dergisi*. 6.3, 85-104.
- Jaeger, H. (2001). "The "Echo State" Approach to Analysing and Training Recurrent Neural Networks". *Technical Report GMD Report 148*. German National Research Center for Information Technology.
- Jaeger, H. (2002). "Tutorial on Training Recurrent Neural Networks, Covering BPPT, RTRL, EKF and the "Echo State Network" approach". *GMD Report 159*. German National Research Center for Information Technology.
- Jagannath, A., J. Jagannath ve P. S. P. V. Kumar. (2022). A Comprehensive Survey on Radio Frequency (rf) Fingerprinting: Traditional Approaches, Deep Learning, and Open Challenges. *arXiv preprint arXiv:2201.00680*.
- Jakhar, D., ve I. Kaur. (2020). Artificial Intelligence, Machine Learning and Deep Learning: Definitions and Differences. *Clinical and Experimental Dermatology*. 45.1, 131-132.
- Jearanaitanakij, K., ve B. Passaya. (2019, October). "Predicting Short Trend of Stocks by Using Convolutional Neural Network and Candlestick Patterns". In *2019 4th International Conference on Information Technology (InCIT)* (s. 159-162). IEEE.

- Jones, J. P., ve L. A. Palmer. (1987). An Evaluation of the Two-Dimensional Gabor filter Model of Simple Receptive Fields in Cat Striate Cortex. *Journal of Neurophysiology*, 58.6, 1233-1258.
- Jordan, J. (2017). Convolutional Neural Networks. <https://www.jeremyjordan.me/convolutional-neural-networks/> (21 Ağustos 2022).
- Jordan, M. L. (1986). “Attractor Dynamics And Parallelism in a Connectionist Sequential Machine”. In: *Proceedings of the Eighth Annual Conference of the Cognitive Science Society* (s. 531-546). Hillsdale, Earlbaum.
- Karakuş, B. A. (2018). Derin Sinir Ağları için Aktivasyon Fonksiyonları. <http://buyukveri.firat.edu.tr/2018/04/17/derin-sinir-aglari-icin-aktivasyon-fonksiyonlari/> (05 Mayıs 2021).
- Kaya, K., ve Ş. Gündüz Ögüdücü. (2020). Deep Flexible Sequential (DFS) Model for Air Pollution Forecasting. *Scientific Reports*. 10.1, 1-12.
- Khurana, K. (2019). Activation Functions in Neural Networks. <https://medium.com/@kshitijkhurana3010/activation-functions-in-neural-networks-ed88c56b611b> (12 Nisan 2021).
- Kızılkaya, Y. M., ve A. Oğuzlar. (2018). Bazı Denetimli Öğrenme Algoritmalarının R Programlama Dili ile Kıyaslanması. *Karadeniz Uluslararası Bilimsel Dergi*. 37.37, 90-98.
- Kızrak, A. (2019). Derin Öğrenme İçin Aktivasyon Fonksiyonlarının Karşılaştırılması. <https://ayyucekizrak.medium.com/derin-%C3%B6%C4%9Fre-nme-i%C3%A7in-aktivasyon-fonksiyonlar%C4%B1n%C4%B1n-kar%C5%9F%C4%B1la%C5%9Ft%C4%B1r%C4%B1lmas%C4%B1-cee17fd1d9cd> (13 Nisan 2021).
- Kim, T., ve H. Y. Kim. (2019). Forecasting Stock Prices With a Feature Fusion LSTM-CNN Model Using Different Representations of the Same Data. *PloS One*. 14.2, e0212320.
- Kingma, D. P., ve J. L. Ba. (2015). “Adam: A Method for Stochastic Optimization”. *International Conference on Learning Representations*, 1-13.
- Kohonen, T., G. Barna ve R. Chrisley. (1988) Statistical Pattern Recognition With Neural Networks: Benchmarking Studies. In: *IEEE International Conference on Neural Networks* (s. 61-68). Vol.1.
- Korkmaz, T. (2013). “Portföy Oluşturmada Temel ve Teknik Analizin Kullanılması”. *Portföy Yönetimi* (s. 146-177). M. Başar (Ed.). Eskişehir: Anadolu Üniversitesi.

- Krizhevsky, A., I. Sutskever ve G. E. Hinton. (2012). "Imagenet Classification With Deep Convolutional Neural Networks". *Advances in Neural Information Processing Systems*. F. Pereira, C. J. Burges, L. Bottou ve K. Q. Weinberger. (Ed.). 25 (NIPS 2012).
- Krus, D. J., ve E. A. Fuller. (1982). Computer-Assisted Multicross-Validation in Regression Analysis. *Educational and Psychological Measurement*. 42, 187-193.
- Kurtz, A. K. (1948). A Research Test of The Rorschach Test. *Personnel Psychology*. 1, 41-53
- Kusuma, R. M. I., T. T. Ho, W. C. Kao, Y. Y. Ou ve K. L. Hua. (2019). Using Deep Learning Neural Networks and Candlestick Chart Representation to Predict Stock Market. *arXiv preprint arXiv:1903.12258*.
- Landahl, H. D., W. S. McCulloch ve W. Pitts (1943). A Statistical Consequence of The Logical Calculus of Nervous Nets. *Bulletin of Mathematical Biophysics*. 5. 135-137.
- Lang, K. J., A. H. Waibel ve G. E. Hinton. (1990). A Time-delay Neural Network Architecture for Isolated Word Recognition. *Neural Networks*. 3.1, 23-43
- Lang, K. J., ve G. E. Hinton. (1988). The Development of The Time-Delay Neural Network Architecture for Speech Recognition. *Technical Report CMU-CS-11-152, Department of Computer Science*. Carnegie Mellon University
- LeCun, Y., B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard ve L. D. Jackel. (1989). Backpropagation Applied oo Handwritten Zip Code Recognition. *Neural Computation*. 1.4, 541-551.
- LeCun, Y., L. Bottou, Y. Bengio ve P. Haffner. (1998). Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*. 86.11, 2278-2324.
- LeCun, Y., Y. Bengio ve G. Hinton. (2015). Deep Learning. *Nature*. 521.7553, 436-444.
- Lewis, N. D. (2016). *Deep Time Series Forecasting With Python*. Create Space Independent Publishing Platform.
- Li, Y., P. Liu ve Z. Wang. (2022). Stock Trading Strategies Based on Deep Reinforcement Learning. *Scientific Programming*. 1-15. <https://doi.org/10.1155/2022/4698656>
- Lipton, Z. C., J. Berkowitz ve C. Elkan (2015). A Critical Review of Recurrent Neural Networks for Sequence Learning. *arXiv preprint arXiv:1506.00019*.
- Liu, H., L. Qi ve M. Sun. (2022). Short-Term Stock Price Prediction Based on CAE-LSTM Method. *Wireless Communications and Mobile Computing*. (S1), 1-7. <http://doi.org/10.1155/2022/4809632>

- Liu, P., Y. Zhang, F. Bao, X. Yao ve C. Zhang. (2023). Multi-Type Data Fusion Framework Based on Deep Reinforcement Learning for Algorithmic Trading. *Applied Intelligence*. 53.2, 1683-1706.
- McCulloch, W. S., ve W. Pitts (1943). A logical Calculus of Ideas Immanent in Nervous Activity. *Bulletin of Mathematical Biophysics*. 5, 115-133.
- Mehrotra, K., C. K. Mohanve R. Ranka. (1996). *Elements of Artificial Neural Networks*. USA: MIT Press.
- Minsky, M. L., ve S. A. Papert. (1969). *Perceptrons*. MIT Press, Cambridge.
- Morris, G. (2006). *Candlestick Charting Explained: Timeless Techniques for Trading Stocks and Futures*. Third Edition, McGraw-Hill. ABD.
- Mosier, C. (1951). Problems and Designs of Cross-Validation. *Educational and Psychological Measurement*. 11, 5-11.
- Mosser, L., O. Dubruleve ve M. J. Blunt. (2017). Stochastic Reconstruction of an Oolitic Limestone by Generative Adversarial Networks. *Transp Porous Media*. 125, 81-103.
- Mousa, R., S. Nazari, A. K. Abadi, R. Shoukhcheshm, M. N. Pirzadeh ve L. Safari. (2021). TI-Capsule: Capsule Network for Stock Exchange Prediction. *arXiv preprint arXiv:2102.07718*.
- Mulla, M. Z. (2020). Cost, Activation, Loss Function|| Neural Network|| Deep Learning. What Are These? <https://medium.com/@zeeshanmulla/cost-activation-loss-function-neural-network-deep-learning-what-are-these-91167825a4de> (20 Haziran 2021).
- Murphy, J. J. (1999). *Study Guide to Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications*. Penguin.
- Nabiyev, V. V. (2021). *Yapay Zeka Derin Öğrenme – Stratejili Oyunlar – Örüntü Tanıma – Doğal Dil İşleme*. İstanbul: Seçkin Yayıncılık.
- Najafabadi, M. M., F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald ve E. Muharemagic. (2015). Deep Learning Applications and Challenges in Big Data analytics. *J. Big Data*. 2.1.
- Nielsen, M. A. (2015). *Neural Networks and Deep Learning*. (Vol. 25). San Francisco, CA, USA: Determination Press.
- Nison, S. (2003). *The Candlestick Course*. USA: John Wiley & Sons.
- Olshausen, B. A., ve Field, D. J. (1996). Emergence of Simple-cell Receptive Field Properties by Learning A Sparse Code for Natural Images. *Nature*. 381.6583, 607-609.

- Oualil, Y., M. Singh, C. Greenberg ve D. Klakow. (2017). Long-Short Range Context Neural Networks for Language Modeling. *arXiv preprint arXiv:1708.06555*.
- Pan, W., J. Li ve X. Li. (2020). Portfolio Learning Based on Deep Learning. *Future Internet*. 12.11, 202.
- Parmar, R. (2018). Common Loss Functions in Machine Learning: <https://towardsdatascience.com/common-loss-functions-in-machine-learning-46af0ffc4d23> (07 Haziran 2021).
- Patterson, J., ve A. Gibson. (2017). *Deep Learning: A Practitioner's Approach*. USA. O'Reilly Media, Inc.
- Quan, Z. Y. (2013, April). "Stock Prediction by Searching Similar Candlestick Charts". In *2013 IEEE 29th International Conference on Data Engineering Workshops (ICDEW)* (s. 322-325). IEEE.
- Ramachandran, P., B. Zoph ve Q. V. Le. (2017). Searching for Activation Functions. *arXiv preprint arXiv:1710.05941*.
- Raschka, S., ve V. Mirjalili. (2017). *Machine Learning and Deep Learning with Python, scikit-learn and Tensor Flow*. Birmingham: Packt Publishing.
- Rashevsky, N. (1938). *Mathematical Biophysics*. Chicago: University of Chicago Press.
- Roffo, G. (2017). Ranking to Learn and Learning to Rank: On The Role of Ranking in Pattern Recognition Applications. *arXiv preprint arXiv:1706.05933*.
- Rosenblatt, F. (1958). The Perceptron, A Probabilistic Model for Information Storage and Organization in the Brain. *Psych. Review*. 62.386, 108.
- Ruder, S. (2016). An Overview of Gradient Descent Optimization Algorithms. *arXiv, abs/1609.04747*.
- Rumelhart, D. E., G. E. Hinton ve R. J. Williams. (1986). Learning Representations By Back-Propagating Errors. *Nature*. 323.6088, 533-536.
- Sakınç, S. Ö. (2018). Hisse Senedine Yatırım Yaparken Finansal Analiz ile Teknik Analiz Yöntemlerinin Birlikte Kullanılmasının Önemi ve Bist'de Bir Uygulama. *Sosyal Araştırmalar ve Davranış Bilimleri Dergisi*. 4.5, 134-153.
- Santur, Y. (2022). Candlestick Chart Based Trading System Using Ensemble Learning For Financial Assets. *Sigma Journal of Engineering and Natural Sciences*. 40.2, 370-379.
- Sattarov, O., A. Muminov, C. W. Lee, H. K. Kang, R. Oh, J. Ahn ve H. S. Jeon. (2020). Recommending Cryptocurrency Trading Points With Deep Reinforcement Learning Approach. *Applied Sciences*. 10.4, 1506.

- Schmidhuber, J. (2015). Deep Learning in Neural Networks: An Overview. *Neural Network*. 61, 85-117.
- Seyyarer, E., F. Ayata, T. Uçkan ve A. Karcı. (2020). Derin Öğrenmede Kullanılan Optimizasyon Algoritmalarının Uygulanması ve Kıyaslanması. *Computer Science*. 5.2, 90-98.
- Sezer, O. B., ve A. M. Ozbayoglu. (2018). Algorithmic Financial Trading With Deep Convolutional Neural Networks: Time Series to Image Conversion Approach. *Applied Soft Computing*. 70, 525-538.
- Sezer, O. B., ve A. M. Ozbayoglu. (2019). Financial Trading Model With Stock Bar Chart Image Time Series With Deep Convolutional Neural Networks. *arXiv preprint arXiv:1903.04610*.
- Sharma, A. (2017). Understanding Activation Functions in Deep Learning. <https://www.learnopencv.com/understanding-activation-functions-in-deep-learning/> (25 Nisan 2021)
- Sharma, M., ve M. Malik. (2021). Use of RNN in Devangari Script. In: *Intelligent Computing and Applications. Advances in Intelligent Systems and Computing*, S. S. Dash, S. Das ve B. K. Panigrahi (Eds.). vol.1172. Singapore: Springer.
- Shen, D., G. Wu ve H. I. Suk. (2017). Deep Learning in Medical Image Analysis. *Annual Review of Biomedical Engineering*. 19, 221.
- Shyam, R. (2021). Convolutional Neural Network and its Architectures. *Journal of Computer Technology & Applications*. 12.2, 6-14.
- Siarni-Namini, S., N. Tavakoli ve A. S. Namin. (2019). A Comparative Analysis of Forecasting Financial Time Series Using Arima, lstm, and Bilstm. *arXiv preprint arXiv:1911.09512*.
- Simonyan, K., ve A. Zisserman. (2014). Very Deep Convolutional Networks For Large-Scale Image Recognition. *arXiv preprint arXiv:1409.1556*.
- Singh, K., G. Gupta, L. Vig, G. Shroff ve P. Agarwal. (2017). Deep Convolutional Neural Networks for Pairwise Causality. *arXiv, arXiv:1701.00597*.
- Singh, P., P. Raj ve V. P. Namboodiri. (2020). EDS Pooling Layer. *Image and Vision Computing*. 98, 103923.
- Skansi, S. (2018). *Introduction to Deep Learning: From Logical Calculus to Artificial Intelligence*. Springer.
- Sosnovik, I., M. Szmaja ve A. Smeulders. (2020). Scale-Equivariant Steerable Networks. *arXiv preprint arXiv:1910.11093*.

- Suresh. A. S. (2013). A Study on Fundamental and Technical Analysis. *International Journal of Marketing, Financial Services & Management Research*. 2.5, 44-59.
- Szegedy, C., W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke ve A. Rabinovich. (2015). Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (s. 1-9). IEEE.
- Talathi, S. S., ve A. Vartak. (2016). Improving Performance of Recurrent Neural Network With Relu Nonlinearity. *arXiv preprint arXiv:1511.03771*.
- Talleg, C., ve Y. Ollivier. (2017). Unbiasing Truncated Backpropagation Through Time. *arXiv preprint arXiv:1705.08209*.
- Taylor, W. K. (1956). Electrical Simulation of Some Nervous System Functional Activities. *Information Science*. 314-328.
- Temel, G. O., S. Erdoğan ve H. Ankaralı (2012). Sınıflama Modelinin Performansını Değerlendirmede Yeniden Örnekleme Yöntemlerinin Kullanımı. *Bilişim Teknolojileri Dergisi*. 5.3, 1-8.
- Tibshirani, R. (1996). Regression Shrinkage And Selection Via The Lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1), 267-288.
- Toussaint, M. (2018). Introduction to Optimization, Second Order Optimization Methods. <https://www.user.tu-berlin.de/mtoussai/teaching/13-Optimization/04-secondOrderOpt.pdf> (01 Temmuz 2023)
- Tsai, C. F., ve Z. Y. Quan. (2014). Stock prediction by searching for similarities in candlestick charts. *ACM Transactions on Management Information Systems (TMIS)*, 5.2, 1-21.
- TSPAKB. (2011). *Analiz Yöntemleri Sermaye Piyasası Faaliyetleri İleri Düzey Lisansı Eğitimi*, Ankara: TSPAKB.
- Van Essen, B., H. Kim, R. Pearce, K. Boakye ve B. Chen. (2015). LBANN: Livermore Big Artificial Neural Network HPC Toolkit. In *Proceedings of the Workshop on Machine Learning in High-Performance Computing Environments* (s. 1-6). <https://doi.org/10.1145/2834892.2834897>
- Velay, M., ve F. Daniel. (2018). Stock Chart Pattern Recognition With Deep Learning. *arXiv preprint arXiv:1808.00418*.
- Wang, Q., Y. Ma, K. Zhao ve Y. Tian. (2020). A Comprehensive Survey of Loss Functions in Machine Learning. *Annals of Data Science*, 1-26.
- Wang, Y., L. Wang, F. Yang, W. Di ve Q. Chang. (2021). Advantages of Direct Input-to-Output Connections in Neural Networks: The Elman Network for Stock Index Forecasting. *Information Sciences*. 547, 1066-1079.

- Weinberger, K. (2015). CS4780/CS5780: Machine Learning for Intelligent Systems. <http://www.cs.cornell.edu/courses/cs4780/2015fa/web/lecturenotes/lecturenote10.html> (11 Mart 2022).
- Werbos, P. J. (1990). Backpropagation Through Time: What it Does and How to do it. *Proceedings of the IEEE*. 78.10, 1550-1560.
- Widrow, B. ve M. Hoff. (1960). Adaptive Switching Circuits. In *Western Electronic Show and Convention, Convention Record* (96-104). vol.4, Institute of Radio Engineers (now IEEE).
- Wiener, N. (1948). *Cybernetics, or Control and Communication in the Animal and the Machine*. New York: John Wiley & Sons.
- Wu, Y., J. Li, Y. Kong ve Y. Fu. (2016, October). Deep Convolutional Neural Network With Independent Softmax for Large Scale Face Recognition. In *Proceedings of the 24th ACM international conference on Multimedia* (s. 1063-1067). <http://doi.org/10.1145/2964284.2984060>
- Xu, B., R. Huang ve M. Li. (2016). Revise Saturated Activation Functions. *arXiv preprint arXiv:1602.05980*.
- Yamashita, R., M. Nishio, R. K. G. Do ve K. Togashi. (2018). Convolutional Neural Networks: An Overview and Application in Radiology. *Insights Into Imaging*. 9.4, 611-629.
- Yılmaz, A., ve U. Kaya. (2019). *Derin Öğrenme*. İstanbul: KODLAB Yayın Dağıtım.
- Zhang, A., Z. C. Lipton, M. Li ve A. J. Smola. (2021). Dive Into Deep Learning. *arXiv preprint arXiv:2106.11342*.
- Zhou, Z.-H., N. V. Chawla, Y. Jin ve G. L. Williams. (2014). Big Data Opportunities and Challenges: Discussions From Data Analytics Perspectives. *IEEE Computational Intelligence Magazine*. 9, 62-74.